Evaluation of Deep Learning Models for Remote Sensing Segmentation and Classification

SaiVenkataLakshmi Ananth
Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, India
saiphd91@gmail.com

Suryakanth Gangashetty
Department of Computer Science and Engineering
Koneru Lakshmaiah Education Foundation, India
svg@kluniversity.in

Abstract: *The rapid changes in Deep Learning (DL) have made better performing models for Remote Sensing Images* (*RSIs), particularly for semantic segmentation and multi-class classification. This study looks at how two common DL structures, U-Net and DeepLabV3+, perform on segmentation tasks, while 201-layer Densely Connected Convolutional Network (DenseNet201-CNN) is compared to Visual Geometry Group 16-layer network (VGG16) for classification tasks. The dataset for segmentation has aerial images of Dubai labeled with pixel-level segmentation across six classes: Building, land, road, vegetation, water, and unlabeled. The classification data set called Remote Sensing Image-Collection of Benchmark 256 (RSI-CB256) has Remote Sensing (RS) pictures sorted into four groups: Cloudy, desert, green_area, and water. DeepLabV3+ demonstrated better training performance and convergence behavior compared to U-Net, exhibiting more stable learning and efficient boundary detection during segmentation. While both models performed competitively, DeepLabV3+ consistently showed stronger generalization capability, making it more effective in delineating complex land cover boundaries. In contrast, U-Net displayed sensitivity to hyperparameters and greater variation in performance, indicating the need for further tuning and regularization. U-Net was good initially but had varied performance and was sensitive to hyperparameters suggesting the need of better regularization techniques. In terms of the classification, DenseNet201-CNN did better than VGG16 in precision, recall, and F1-score for all categories. Notable performance gains were observed in "cloudy" and "desert" classes where DenseNet201-CNN model demonstrated significantly fewer misclassifications. Overall, DenseNet201-CNN outperformed VGG16 in terms of total classification accuracy. These results establish DenseNet201-CNN as a superior choice in RSI classification tasks in this study*.

Keywords: *Semantic segmentation, deep learning, VGG16, classification, remote sensing, densenet201-CNN, U-net, deeplabV3+*.

Received April 24, 2025; accepted July 18, 2025
https://doi.org/10.34028/iajit/22/6/7

1. Introduction

Remote Sensing Image (RSI) segmentation and classification made big progress with Deep Learning (DL). Traditional segmentation methods often struggle with high-resolution images, this has open doors for researchers to look for better DL methods.

To address low accuracy and extracting relevant features form RSIs, Bilgin *et al*. [3] proposed cluster validation technique utilizing a One-Class Support Vector Machine (OC-SVM) and a new subtractive-clustering-based similarity segmentation are presented for unsupervised hyperspectral image segmentation. The proposed Power of the Support Vector-Spectral Discrimination (SV-PWSD) method is a new kind of hyperspectral image segmentation cluster validity index, which is based on the PWSD measure and used to identify the appropriate number of groups in the input area.

Another study on DL methods [32] looks at high-resolution RSIs from Gaofen-2 Satellite (GF-2) by using U-Net for binary and multi-class classification. The dataset consists of six types of features, including buildings, land, water, grass, and forests. A new method called "environmental voting" figures out the category for uncertain pixels by looking at the differences between similarity and contrast. The U-Net neural network reached an overall classification accuracy of 93.83%.

In the direction of refining the segmentation further, this paper [29] talks about making segmentation methods better. It presents Communication and Attention Segmentation-Network (CAS-Net), which is a kind of communication network for Remote Sensing (RS). This new model swaps out the cascade convolution in feature extraction with something called Spatial Pyramid Dilated Convolution (SPD-Conv) convolution and adds a pooling layer to keep important details from getting lost. This change helps make small object segmentation much better. Additionally, there's the Atrous Spatial Pyramid Pooling (ASPP) module that works with CA to boost how well objects are recognized and targeted in RSIs. When tested against other top models on the International Society for Photogrammetry and Remote Sensing (ISPRS) Vaihingen dataset, CAS-Net shows it performs better by fixing problems related to small object segmentation and uneven grouping in datasets.

To achieve a better accuracy and also maintaining

high speed in segmentation process, the swin transformer introduces an efficient transformer-based approach [28]. This method employs explicit and implicit edge enhancement techniques to tackle issues with edge objects. Compared to High-Resolution Context Network_Width 48 (HRCNet_W48) on the Potsdam dataset, the proposed method improves accuracy and balances computational complexity (flops) with segmentation performance. Experimental results show a 3.23% improvement in accuracy on Vaihingen and a 2% increase in mean Intersection over Union (mIoU).

Keeping the above work in mind in terms of better metrics, research gaps and areas to enhance further, our method uses U-Net and DeepLabV3+, which is based on the flexibility of segmentation models. It can fit to various datasets with small changes. In contrast to other works that are dataset-specific, these models perform well with many kinds of RSIs. This cuts down the need for manual adjustments while still keeping accurate segmentation. This work makes it easy to adjust to different datasets and keeps high accuracy with little manual work needed.

Apart from segmentation, Remote Sensing Image Scene Classification (RSISC) is very important in using things like city planning, watching disasters, and checking the environment. New studies show that DL models do better than older ways of classifying. Pham *et al*. [16] showed a DL RSISC system that uses transfer learning and attention methods with many heads. The model gets 94.7% accuracy when classifying on the Northwestern Polytechnical University Remote Sensing Image Scene Classification 45-Class (NWPU-RESISC45) dataset, showing it is good and can be used in real life situations.

A new model for classification [10] makes image classification techniques better with Denoising Diffusion Probabilistic Model (DDPM). This process takes the spatial features from pairs of Hyper-Spectral Imagery (HSI) and Light Detection And Ranging (LiDAR) data, mixing them with hyperspectral traits for joint training. Tests show that using middle activation in the reverse diffusion phase boosts how precise the classification is, even when adding multispectral and Synthetic Aperture Radar (SAR) data.

Still, even with progress, the RS field uses small datasets that depend on applications, which creates problems for model benchmarking with large-scale datasets common in general computer vision tasks. To fix this issue, Michael and Wu [13] made a version of the Sentinel 1 and 2 Multi-Spectral datasets (SEN12MS) focused on classification. They tested several basic models using regular Convolutional Neural Network (CNN) structures and looked into merging multi-spectral and multi-sensor data. The findings support that combining different data types is better than just using RGB images for classifying RSIs.

A meta-analysis of existing studies [22] categorizes DL-based RSI classification methods into three primary groups: architectures based on CNNs, Vision Transformers (ViTs), and Generative Adversarial Networks (GANs). The analysis identifies Aerial Image Dataset (AID) and NWPU-RESISC45 as the most frequently used datasets, highlighting a paradigm shift towards transformer-based models since 2021. The study also reviews challenges and future research directions in RS classification.

Considering the adaptability challenges of dataset-specific models, 201-layer Densely Connected Convolutional Network (DenseNet201-CNN) and Visual Geometry Group 16-layer network (VGG16) demonstrate robust generalization across multiple RS datasets with minimal adjustments. Prior studies highlight the limitations of using small, dataset-specific models that hinder generalization. Our proposed models mitigate these issues by providing a flexible framework that adapts to new datasets with minimal tuning, ensuring broad applicability in RS segmentation and classification.

This study develops and applies DL models for RS analysis. The implemented methods are U-Net and DeepLabV3+ for segmentation, and DenseNet201-CNN and VGG16 for multi class classification to handle good quality RSIs. These models help in clearly showing types of land cover and finding objects, useful for city planning, watching the environment, and handling disasters. By putting together strong feature picking methods, these models get high correct classification on many datasets, making it easy to grow bigger or change for other RS uses.

The article is structured in the following manner: Section 2 outlines the related work. Section 3 has proposed implementation flow. In section 4, there is a computational analysis and presents the findings. Finally, section 6 encompasses the conclusion and offers future paths for this research.

2. Related Work

Semantic segmentation of RSI is extensively utilized in geological research, urban management, and disaster monitoring. Recent solutions for remote segmentation jobs are typically addressed using CNN-based and transformer-based models. These are extensively utilized in geological research, urban management, and disaster monitoring. In particular, transformer-based architectures often face two major problems: high performance demands and misclassification. Therefore, Xu et al. [28] proposed a new transformer model to distribute competitive advantage in order to overcome these issues. First, a pure efficient transformer with mlphead is built to boost the transformer's speed based on the Swin Transformer backbone. To address the issues with edge objects, both explicit and implicit edge enhancement techniques are also suggested. Results from studies performed on the Potsdam and Vaihingen

datasets indicate that the suggested approach increases final accuracy and strikes a balance between accuracy (Efficient-L) and computational complexity (flops), achieving 2.6% mIoU on Vaihingen and 3.23% of Vaihingen improvement (in comparison to HRCNet_W48 on Potsdam). Consequently, it is thought that the need for high-performance switching will help address the issues with RS picture segmentation.

The significance of disregarding correlations is widely recognized in the fields of statistics and machine learning. As data volume escalates, numerous unsupervised categorization algorithms grow more intricate. The K-means clustering approach is prevalent and extensively utilized in the machine learning community owing to its simplicity and efficacy in clustering substantial datasets. Nonetheless, as to other integration techniques, it necessitates the prior selection of equivalent groupings. Ali *et al*. [1] focus on developing a new method for finding the best fit of k clusters using a data-driven approach. Considering the symmetry of the cluster, the K-means algorithm is usually used for a range of k values, which should be equal to the best k value. Ali *et al*. [1] selected the final k value as the consensus value based on the distinctiveness and similarity of the cluster findings' center values. Using real data from the UCI machine learning library, Ali *et al*. [1] tested the suggested method's performance on several simulated datasets with control settings. They also use satellite photos of the Islamabad, Pakistan region captured by the U.S. Geological Survey's Sentinel-2B satellite to assess the effectiveness of the suggested approach for RS (such as urbanization and deforestation). It is evident from testing the outcomes and examining the actual data that the suggested approach outperforms the conventional technique in terms of accuracy and root mean square error.

Since agricultural lands are the foundation of national agriculture, obtaining information about agricultural land distribution is beneficial for further monitoring of agricultural production. However, by combining DL techniques to analyze remote images, previous studies could not sufficiently compensate for the uncertainty of the boundary, so the reality is that there is little in agricultural inference and needs to be improved. Pan *et al*. [15] analyzed the pertinent literature and employs the K-means model, U-Net model, and DeelLabV3 model to arrange and rectify RSIs of agricultural land inference models in order to solve the drawbacks of current approaches and increase overall accuracy. Following model training and parameter assessment, the average percentage of agricultural lands is 85.44%.

Deep neural networks are effective in extracting roads from high-resolution satellite images. A network with scalability will reveal various and interesting ways for the network to benefit. Tao *et al*. [21] developed a spatially aware intelligence architecture that can facilitate multi-pixel message delivery when embedded in a traditional semantic segmentation framework. The road representation approach can learn both local road aspects and global road information since representation techniques can be used to present and improve spatial data. The road's (i.e., continuous and changeable road) spatial reaction. Thus, this approach can successfully address the issue of congestion while guaranteeing the shooting process's continuity. Three sizable Very High Resolution (VHR) satellite imaging datasets were used for validation, and the results demonstrate that the suggested approach will increase extraction accuracy and efficiency to human standards.

In deep neural networks, spatial pyramid pooling modules or encoder-decoder models are used for semantic segmentation tasks. The original network could encode multiple aspects of the content by analyzing features using filters or joint operations of multiple values and multiple views, whereas the latter network can capture additional information content by gradually recovering clear data boundaries. Chen *et al*. [4] integrated the best outcomes from both methods. Specifically, the proposed DeepLabv3+ model enhances DeepLabv3 by integrating a simple yet effective decoder module to improve segmentation results, especially in object regions. In order to build a more resilient and potent encoder-decoder network, authors continue to examine the Xception model and implement distinctive characteristics in the ASPP and decoder modules. Without any post-processing, they show how well the suggested model performs on the Pattern Analysis, Statistical Modeling and Computational Learning Visual Object Classes (PASCAL VOC) 2012 and cityscapes datasets, obtaining 89% and 82.1% test set success, respectively.

Zhang *et al*. [31] delineated the differentiation between fuzzy clustering methodologies and provides two novel integration techniques. The initial method is referred to as Deviation Sparse Fuzzy C-Means (DSFCM). When faced with spatial relationships, a second approach is proposed, namely the fuzzy C-means method with restricted surrounding data Deep-Deviation Sparse Fuzzy C-Means-Normalized (DSFCM_N). This study makes three contributions. First, the clustering procedure makes advantage of the data's theoretical significance as determined by the significance test. Compared to the fuzzy C language, this can produce cluster locations that are more accurate. Second, DSFCM and DSFCM_N can detect output and noise by differentiating between measured and theoretical values.

Lastly, when the surrounding data is taken into consideration rather than just the data, the estimates of the difference between the observed and theoretical data values are more trustworthy. DSFCM_N performs better than other fuzzy clustering techniques, is more effective, and is therefore more competitive, according to experiments conducted on both simulated and real images.

In order to preserve the observed integration patterns through a thresholding operation, Maldonado *et al*. [12] suggested an unsupervised method that use an embedded notion to choose the most significant features. The premise behind the first group is to penalize the utilization of the measuring system's features while also lowering the model's inaccuracy. Their foundational method is kernel K-means, which functions similarly to one of the most widely used clustering algorithms, K-means, but provides a great deal more flexibility. Once more, the discovery of many clusters is a result of the distance calculation using kernel functions. They experiment with numerous datasets and suggest an approach to address the associated minimization problem.

Semantic segmentation of RSIs is a critical concern in numerous applications. Conventional encoder-decoders utilizing CNNs employ cascade pooling processes to aggregate semantic information, leading to a decline in localization precision and the retention of spatial content. Zhang *et al*. [30] presented the High-Resolution Network (HRNet), which generates high-resolution images without necessitating phase separation to circumvent these limits. In order to improve the integration of context-related data, authors also improve the low-high features that were taken from various branches. Low-resolution photographs have been used to simulate long-term connections because they are modest and carry a lot of semantic information. By adding the Adaptive Spatial Pooling (ASP) module, which pools several local points, the maximum resolution is increased.

Combining aggregated data produced at several levels yields outcomes that can take advantage of geographical settings at both the global and regional levels. This method achieves good state-of-the-art performance and enhances the accuracy of widely used CNNs, according to experimental results on two RSI datasets.

To better utilize explanation models, Ronneberger *et al*. [20] offered a network and training approach based on reliable data. The architecture has a thorough method for achieving accuracy and a compact method for capturing details.

Ronneberger *et al*. [20] showed that such a network can be trained end-to-end with a limited number of images and outperformed the previous best solution (sliding Window) in the International Symposium on Biomedical Imaging (ISBI) competition for segmenting neural structures in electron microscope stacks. In the 2015 ISBI cell tracking challenge, we won by a significant margin in the phase contrast and Differential Interference Contrast (DIC) categories using the same network that we had learned on electron microscope images. The internet is also quick. A 512x512 image can be segmented in less than two seconds on modern Graphics Processing Units (GPUs).

Even in challenging circumstances, humans are able to locate significant areas. This study served as inspiration for the introduction of observation techniques into computer vision, which aimed to replicate this feature of human eyesight. It is possible to think of this tracking as altering the weights based on the input images' characteristics.

Numerous visualization tasks, including image classification, object recognition, segmentation, video comprehension, image production, 3D visualization and multitasking, have been effectively completed using the tracking technique. Numerous computer-based tracking techniques, including channel tracking, spatial tracking, trunk tracking, and branch tracking, are thoroughly analyzed and categorized by Guo *et al*. [7].

Ren *et al*. [19] developed a DL model to classify sea ice and open water from SAR images. The back-end model utilized is U-Net, a prominent Fully Convolutional Network (FCN) designed for pixel-level segmentation.

The DL-based feature extraction model Residual Network (ResNet-34) is utilized as the encoder in U-Net. To achieve high classification, they construct the Dual Supervision U-Net (DAU) model by integrating the binary supervision techniques into the original U-Net to improve the best representation. SAR images are acquired by Sentinel-1A.

The input models are the SAR image's characteristics and the dual polarization data. They train the model using 15 dual polarization images collected close to the Bering Sea, and we test the model using three more images.

Experiments demonstrate that DAU-Net can classify pixels; on three test images, its interconnection efficiency IoU is enhanced by 7.48%, 0.96%, and 0.83%, respectively, in comparison to the original U-Net. The I/O importances of the three DAU-Net components are enhanced by 3.04 percentage points, 2.53 percentage points, and 2.26 percentage points, respectively, in comparison to the recently released DenseNet FCN model.

High-quality images enhance the accuracy and resolution of interpretations in agriculture, forestry, urban planning, and environmental monitoring, thereby aiding in the elimination of target information. An instance of RSI fusion involves the integration of low-resolution HSI and LiDAR data.

Currently, the majority of proposed RSI fusion methods concentrate on the fusion of HSI and LiDAR data features, with insufficient emphasis on the spatial distribution relationship between the two. Jiang *et al*. [10] presented an image fusion distribution network based on the DDPM to address this problem. Though its use in image fusion research has not yet been explored, DDPM can learn the classification information of images through training and create new images with the same classification by differentiating across photos.

Consequently, Jiang *et al*. [10] extracted the difference in correlation coefficients from HSI-LiDAR

data sets using DDPM, merge them with HSI hyperspectral features, and then conduct joint training. According to experimental results, good HSI and LiDAR information can be recovered by averaging a specific amount of time across several procedures. This results in a more accurate classification of people. Furthermore, even when the input is modified from SAR and Multi-Image Sensor (MSI) data, the model continues to function well.

DL has garnered heightened interest in the domain of RS owing to its data representation proficiency. DL models have demonstrated efficacy in identifying RS data according to a specified model. However, the ability to analyze univariate data is still limited due to the lack of variance.

In order to get over this restriction, Hong *et al*. [8] provided a straightforward yet powerful multimodal DL framework for face detection and LiDAR data that is based on EndNet. By requiring the successive reconstruction of features from multimodal inputs, EndNet combines multimodal data. In contrast to several other commonly utilized fusion procedures (such as early, intermediate, and late fusion), this innovative design may improve the integration of neurons across structures. The EndNet implementation outperforms state-of-the-art baseline datasets on hyperspectral-LiDAR classification tasks, as demonstrated by extensive testing on two well-known hyperspectral and LiDAR datasets.

Niazmardi *et al*. [14] introduced Multi-Kernel Learning (MKL) by introducing the basic features of different MKL algorithms and analyzing their features in the problem of RSI classification. The classification of various MKL algorithms is presented first, and a few interesting MKL algorithms are suggested for each group. Specifically, RS is added to the MKL method, which has been used in machine learning thus far. Next, a theoretical comparison of the examined MKL algorithms is made from the following perspectives:

1. Complexity of computing; Following the theoretical comparison, various MKL algorithms are compared experimentally from the standpoints of 1) feature fusion problems.
2. Model selection. Some recommendations for the proper selection of the MKL algorithm are provided based on the theoretical and experimental study of the algorithm.

Rasti *et al*. [17] presented a hyperspectral feature extraction method known as Sparse Smoothed Low-Rank Analysis (SSLRA).

Initially, a novel low-rank model for HSIs is introduced, wherein HSI is decomposed into parallel and discrete components. These components are estimated concurrently through the application of a non-convex Constrained Penalized Cost Function (CPCF).

Additionally, the authors note that this novel style of breakdown enhances the classification of HSI. The experiment uses SSLRA for data from Trento (rural) and Houston (urban).

To create the final classification map, the collected features are fed into a random forest classifier or SVM. When compared to the most advanced feature extraction technique, the results validate the increase in classification accuracy.

Villa *et al*. [24] introduced Independent Component Discriminant Analysis (ICDA) as a method for RS applications. ICDA is a nonlinear analytical method that utilizes Bayesian classification rules for the evaluation of signals produced by Integrated Circuits (ICs). This method employs Independent Component Analysis (ICA) to identify a transformation matrix that maximizes the independence of the transformation components. When the data are plotted in independent space, estimates of the variance functions can be calculated more simply than the product of univariate densities.

Each IC's density was determined using a nonparametric kernel density estimator. Lastly, the distribution was subjected to the Bayesian rule. The viability of classifying hyperspectral pictures using ICDA is examined in this research. The authors provide an efficient way to estimate the ideal number and examine the strategy used to manage the independence and the impact of the number of stored ICs on the distribution. The suggested approach is used to evaluate multiple data kinds (urban/agricultural areas, training size, and sensor type) on a variety of hyperspectral photos. The outcomes are contrasted with SVMs, one of the most widely used classifiers for hyperspectral pictures, and the accuracy comparison of the aforementioned techniques is displayed.

DL has gained popularity as a way to analyze and classify RS photos with high performance. The evolution of different computer vision methods in the realm of RS is explained in detail in this work. Tombe *et al*. [23] presented DL techniques and explores their potential as an open-source software framework for artificial intelligence, given the growing amount of RS datasets with different scene interpretations. The opportunities and gaps that the RS community must fill are also covered in this study.

The field of RS focuses on classifying satellite images into several scene types through analysis. Images produced by satellite sensors are employed in computer vision applications such object recognition, position locating, scene labeling, and image segmentation. Low-level, medium-level, and high-level methods are the categories into which a variety of image feature analysis techniques have been developed. These methods [5, 6, 26] place a strong emphasis on supervised learning, unsupervised learning, and human engineering abilities for feature analysis and representation.

Scale-Invariant Feature Transform (SIFT) is a kind of descriptor [11] that makes a vector to show features

in four steps: finding scale-space extrema, locating keypoint features, assigning orientation, and making local image descriptors. SIFT finds candidate points and spots positions that stay the same even when the image changes size. It finds key points by looking at nearby ones and discarding low contrast or poorly located points. The descriptor for the key points is made for orientation, which does not change with image rotation, producing histograms from gradients. The local image descriptor gives parameters like scale, orientation, and location to each key point to create a stable local 2D system.

Transformer deep learn methods like the Excellent Teacher Network Guiding Small Networks (ET-GSNet) and Label-Free Self-Distillation Contrastive with Transformer Architecture (LaST) can pick up long-distance info. But they need datasets and often don't have enough labeled data for working in new areas.

To fix these problems, fresh research paths in geography-aware DL models are popping up. These models [25, 27] mix knowledge and data, looking at local features, physical features, and space features. Main DL methods look at rule-based stuff, semantic networks, object strategies, physical model stuff, and neural network styles.

Research Gaps

Segmentation

- High computational complexity: transformer based and high-resolution models require extensive resources and fail to scale across diverse environments necessitating efficient design
- Noise sensitivity in heterogeneous environments: clustering methods like K-means are sensitive to noise impacting segmentation reliability in complex landscapes

Classification

- Poor generalization: models many times do not work well in different regions, needing learning methods that change for better fitting.

Addressing these gaps is essential to advancing segmentation and classification capabilities in RS applications. In this work, we employed U-Net and DeepLabV3+ for segmentation, alongside DenseNet201-CNN and VGG16 for classification, to address various challenges in RSI analysis.

For segmentation, U-Net and DeepLabV3+ were picked to fix issues with boundary problems and loss of spatial resolution. The U-Net architecture is up and down and has skip connections which retains information better, outlines better and better segmentation. DeepLabV3+ uses ASPP to input information from different scales while its decoder is better for finding the objects. These models collectively work to balance the demand for high-resolution segmentation with the cost of significant computation, by refining both feature extraction and resolution processing.

Scalability and adaptability are major challenges in RS work. Because DeepLabV3+ can acquire features from various receptive fields, it is powerful in numerous meteorological circumstances that are prevalent in mapping out different geographical features. The architecture of U-Net, which allows analysis at varying resolutions, helps the model generalize across a range of landform types, and overcomes classification buffering errors.

DeepLabV3+ is very effective in extracting multiple receptive fields, it does not work in one place. U-Net's Multi-Resolution Architecture makes it a natural fit for multi-resolution imagery so it is able to generalize better to different terrains and use less computational power. U-Net's skip connections and DeepLabV3+'s ASPP also help reduce noise by preserving small features and strengthening weak spots.

In classification, DenseNet201-CNN and VGG16 were chosen to address noise sensitivity and poor generalization.

First, the DenseNet201-CNN model leverages the advantage of spatial feature extraction that could extract spatial features from complex RS data and come up with meaningful spatial features, thus improving classification accuracy. The much deeper VGG16 architecture addresses this, improves pattern recognition, and generalization to other datasets. These models strike a balance between computational efficiency and classification performance, making them suitable for real-time applications.

3. Proposed Framework

3.1. Segmentation

The workflow diagram shown Figure 1 is about image segmentation which uses Dubai Aerial Imagery data.

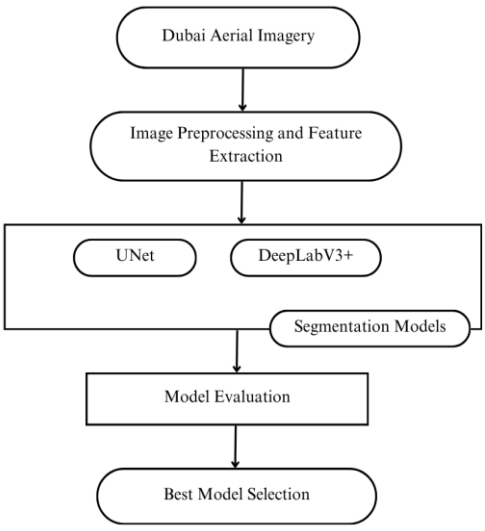


Figure 1. Proposed framework for segmentation model.

The step-by-step procedure includes:

3.2. Dubai Aerial Imagery

It Start with aerial images of Dubai city. These pictures have different land covers like buildings, streets, and plants that must be separated.

Let the input dataset be represented as: $X=\{x_1, x_2, \ldots x_n\}$; $Y=\{y_1, y_2, \ldots y_n\}$, where X consists of aerial images and Y consists of ground truth segmentation masks.

3.2.1. Image Preprocessing and Feature Extraction

This part is about splitting the dataset. Preprocessing has procedures like making them smaller, leveling off their values, and changing them to make the model work better. Finding features helps bring out key bits of the pictures, like outlines and surface patterns that help with splitting.

Each image x_i is preprocessed by normalization and resizing using Equation (1).

$$x_i^1 = (x - \mu) / \sigma \quad (1)$$

where μ and σ are the mean and standard deviation of the dataset.

3.2.2. Segmentation Models

In this workflow, two models for segmentation are employed. The segmentation function f_θ maps input images to output masks as in Equation (2).

$$\hat{Y} = f_\theta(X) \quad (2)$$

Where f_θ represents the DL model with parameters θ. Two architectures used:

1. *U-Net*: this is a well-known architecture in DL made for segmenting images semantically. It has an encoder-decoder setup with connections that skip layers to keep spatial details intact. U-Net follows an encoder-decoder structure with skip connections as in Equation (3)

$$\hat{Y} = Decoder(Encoder(X)) \quad (3)$$

2. *DeepLabV3+*: this is a very advanced model in DL meant for semantic segmentation. It applies Atrous convolutions which help capture different scales of context and give precise maps for segmentation. DeepLabV3+ applies *Atrous* convolutions to capture multi-scale information as in Equation (4),

$$\hat{Y} = AtrousConv(X) \quad (4)$$

- **Model evaluation**: the models that are trained for segmentation get evaluated using different measurement standards like:

  1. Intersection over Union (IoU).
  2. *Pixel Accuracy* (PA).
  3. Dice coefficient.

The predicted segmentation map $\hat{Y}$ is evaluated against ground truth using *IoU* in Equation (5), PA in Equation (6), and *Dice* coefficient in Equation (7).

$$IoU = \Sigma (y_i * \hat{y}_i) / \Sigma (y_i + \hat{y}_i - y_i * \hat{y}_i) \quad (5)$$

$$Pixel\ Accuracy = \Sigma 1(\hat{y}_i = y_i) / N \quad (6)$$

$$Dice = 2\,\Sigma (y_i * \hat{y}_i) / \Sigma\, y_i + \Sigma\, \hat{y}_i \quad (7)$$

For all *i* from 1 to *N*

- **Best model selection**: from the evaluation of metrics, the model that works best is picked for use or to make better. This selection is done by looking at important performance marks like accuracy, *Dice*, *PA* and ability to adapt to new datasets. The model with the highest evaluation metrics (IoU and dice) is selected as in Equation (8).

$$Best\ Model = argmax(IoU, DicePA) \quad (8)$$

These measurements help us in understanding how good the model is at marking various land cover details. In summary, this process seeks to handle aerial pictures for precise segmenting of urban areas and structures using DL schemes.

3.3. Classification

The framework workflow in Figure 2 shows a way to classification images using the Remote Sensing Image-Collection of Benchmark 256 (RSI-CB256) dataset. This includes the following phases.

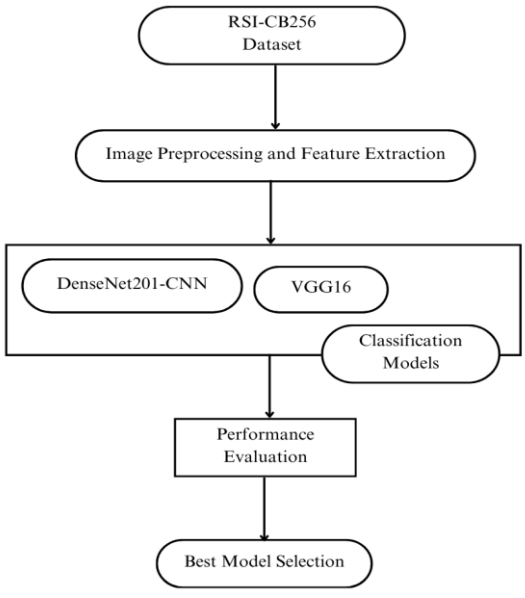


Figure 2. Proposed framework for classification model.

3.3.1. RSI-CB256 Dataset

This framework starts with the RSI-CB256 dataset that has images connected to aerial images (i.e., Looking down from high places). This dataset is used for multi class classification.

3.3.2. Image Preprocessing and Feature Extraction

This section of the framework is about getting images of dataset ready for training the model. In general, this include making them smaller, adjusting values, changing them a bit randomly, and removing unwanted

noise from the images. Also, this phase takes out important details like surfaces, forms, and designs from the images that matter for sorting them correctly.

Given an input image *X* of size H X W X C (e.g., 224 X 224 X 3), preprocessing includes:

3.3.3. Preprocessing Steps

Let *X* represent the raw input image. The preprocessing transformation begins with rescaling where each pixel value gets normalized to the [0,1] using Equation (9):

$$X^1 = X\ /255 \tag{9}$$

Following this, the *VGG16* specific processing is carried out, as it needs input to be scaled according to its specific preprocessing function as in Equation (10).

$$X^1 = VGG16\_preprocess(X) \tag{10}$$

After normalization and model-specific preprocessing, data augmentation is performed to introduce visual variability that enhances model generalization. As represented in Equation (11), augmentation includes operations such as rotation by an angle θ, shifting by a displacement δ, zooming by a scaling factor *z*, and flipping.

$$X^1 = rotation(X,\theta) + shift(X,\delta) + zoom(X,z) + flip(X) \tag{11}$$

Once the augmented and preprocessed images X_1 are ready, feature extraction begins. In the first approach, the VGG16 model with pretrained ImageNet weights acts as a feature extractor. Let *f*(.) represent the feature extractor, as represented in Equation (12).

$$F = f(X^1) \tag{12}$$

The VGG16 model includes 13 convolutional layers and 5 max-pooling layers in the following configuration:

1. **Block 1**: Conv(64) → Conv(64) → MaxPool.
2. **Block 2**: Conv(128) → Conv(128) → MaxPool.
3. **Block 3**: Conv(256) → Conv(256) → Conv(256) → MaxPool.
4. **Block 4**: Conv(512) → Conv(512) → Conv(512) → MaxPool.
5. **Block 5**: Conv(512) → Conv(512) → Conv(512) → MaxPool.

We will use the pre-trained VGG16 model up to the last max-pooling layer and add custom dense layers for classification.

In the case of DenseNet201, for each convolutional layer, the feature maps are computed using the formula represented in Equation (13).

$$Y_{i,j,k} = \Sigma\Sigma\Sigma\ X_{i+m,j+n,c} * F^{(k)}_{m,n,c} + b_k \tag{13}$$

where:

$Y_{i,j,k}$: is the feature map output at position (i,j) for filter k.
X: is the input feature map.
F: is the convolution filter of size f X f, b_k is the bias term.

Each convolution is followed by a ReLU activation function using Equation (14).

$$f(x) = max(0,x) \tag{14}$$

3.4. Model Integration, Evaluation, and Selection Framework

In this section, we describe dimensionality reduction, classification and model evaluation framework. The pooling layer reduces the dimensionality of feature maps, while keeping crucial spatial information. The derived low-level features are subsequently fed to different classification models that learn discriminating characteristics for accurate classifying brain tumor types. Finally, we conduct a comprehensive experimental process and evaluate the considered model in terms of accuracy, precision, recall, and F1-score to make an objective comparison among various models for diagnosis.

3.4.1. Pooling Layer (Dimensionality Reduction)

For All *m*, *n* ranges from 0 to (p-1). This reduces spatial dimensions while retaining important features. After extracting feature maps, the output is flattened into a one-dimensional vector represented using Equation (16).

$$Y_{i,j,k} = max\ X_{i+m,j+n,k} \tag{15}$$

$$X_{flat} = reshape\ (F) \tag{16}$$

Fully connected layers are represented using Equation (17)

$$Y = WX_{flat} + b \tag{17}$$

Using ReLU activation is as in Equation (18).

$$f(Y)\ max(0,Y) \tag{18}$$

3.4.2. Classification Models

This pipeline evaluates two model types for classification:

- *DenseNet201-CNN*: a deep learn model that learns features from images in a way that is spatial.
- *VGG16*: an already trained deep CNN architecture famous for how deep it is and its performance on tasks of image classification.

This work trains these models use the preprocessed data and extract features to find patterns and sort pictures into classes.

The final dense layer outputs four units (corresponding to the four classes), followed by softmax activation:

$$\hat{y}_i = exp(z_i)\ /\ \Sigma\ exp(z_j) \tag{19}$$

where:$\hat{y}_i$represents the predicted probability for class *i*, z_i is the output of the final dense layer. *j* ranges from 1 to 4 of predicted class

The model is compiled using the categorical cross-entropy loss function

$$Ł = -\Sigma\Sigma\, y_{ij}\, log(\hat{y}_{ij}) \quad (20)$$

For All *i* is from 1 to *N* and *j* is from 1 to 4.

Optimizer is Adam with learning rate 0.0001. After training, the model predicts the class of a new input image. In this regard, the hyperparameters used in our experiment, including a learning rate of 0.0001, a batch size of 32, and training for 50 epochs, are aligned with the state of the art in satellite image classification tasks with CNNs. These values, which were selected based on pretesting and literature recommendations, represent a trade-off between convergence rate and stability of the model. The Adam optimizer was also selected, based on its ability to learn adaptively and its demonstrated performance on image-based DL problems.

$$\hat{y} = model(X_{test}) \quad (21)$$

Predicted class is represented by,

$$\hat{C} = argmax(\hat{y}) \quad (22)$$

3.4.3. Evaluating Performance

models get evaluated after training with many measures like accuracy, precision, recall, F1-score, and confusion matrix. This helps to see how good models are on new data not seen before.

3.4.4. Choosing Best Model

From the performance checks done before, select best model for use or more tuning. The model that has highest accuracy and can work well generally is picked for last step of using it.

In general, this process works to handle satellite or aerial images with DL methods to get right classification results and find the best model for real usage.

4. Computational Analysis

4.1. Segmentation Model Performance Analysis

Semantic segmentation is a critical task in computer vision, where models aim to classify each pixel of an image into a predefined category. In this analysis, we compare the performance of two widely used segmentation models, U-Net and DeepLabV3+, by evaluating their training and validation metrics across multiple epochs. The analysis focuses on individual performance insights and a comparative study to determine their strengths and weaknesses.

4.1.1. Dataset

The dataset [9] for the segmentation comes from "Humans in the Loop" collection, with aerial images of Dubai. It has 72 images with 512X512 size that are arranged into 6 big tiles. Each image is marked with pixel-wise segmentation, putting objects into six classes: Buildings, land, roads, vegetation, water, and areas without labels. The marks on pixels give good info to help split up different parts in the aerial photos.

This segmentation set faces some hard stuff like class imbalance and how close together different classes can be such as roads and land which may look alike. Even with these hard things, this dataset helps to make and check better models for segmentation.

4.2. U-Net Model Performance

The U-Net model is a CNN primarily designed for biomedical image segmentation. It consists of an encoder-decoder architecture with skip connections to retain spatial information. The analysis of its training performance reveals the following insights:

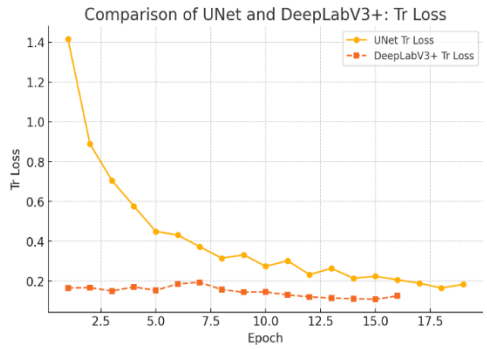


Figure 3. Train loss over epochs U-Net and DeepLabV3+.

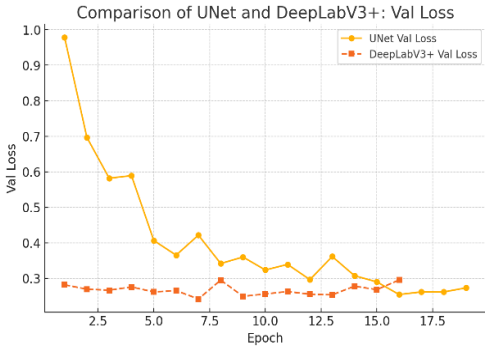


Figure 4. Val loss over epochs U-Net and DeepLabV3+.

The training loss starts at a relatively high value and decreases over epochs, indicating the model is learning effectively as in Figure 3. However, fluctuations in the loss suggest some sensitivity to hyperparameter tuning. The validation loss initially follows a decreasing trend but exhibits fluctuations, which might indicate the model struggles to generalize effectively to unseen data as in Figure 4.

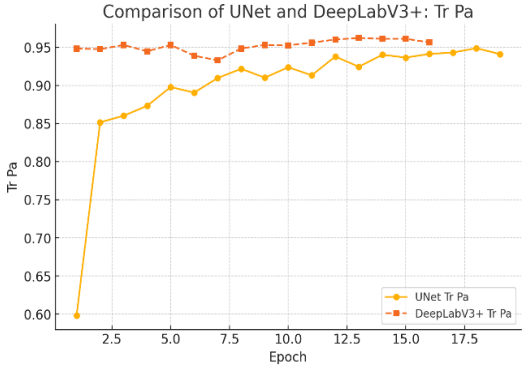


Figure 5. Train PA over epochs U-Net and DeepLabV3+.

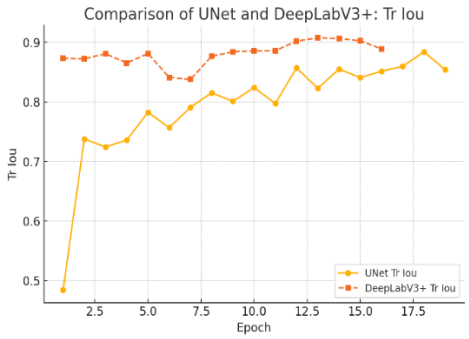


Figure 6. Train IoU over epochs U-Net and DeepLabV3+.

The well-trained model, as observed in training PA in Figure 5 and IoU represented in Figure 6, achieves better performance with each epoch, which is because of the increased ability of the model to capture the segmentation boundary. Though, PA in Figure 7 and the validation IoU in Figure 8 shows fluctuations which suggest that more regularization or augmentation might be required.

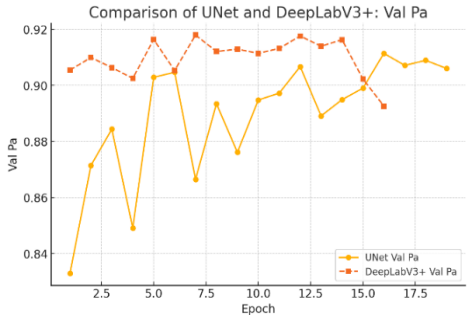


Figure 7. Val PA over epochs U-Net and DeepLabV3+.

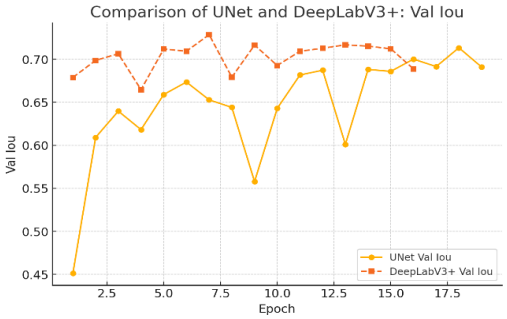


Figure 8. Val IoU over epochs U-Net and DeepLabV3+.

The evolution of Figures 9 and 10 of gradient analysis of U-Net tells how the model learns and optimize. The key observations are:

1. The gradient of the training loss also has a steep incline for the first few epochs indicating that learning happens rather quickly. But as it learns, the fluctuations in the gradients indicate that it is learning at different rates and there are spikes and drops every now and then. This suggests that the model has a hard time with a constant learning rate and may be sensitive to hyperparameter tuning.
2. Validation loss gradient exhibits random spikes. This indicates that the model is finding it hard to generalize to unseen examples. The different behaviors of the gradient means that model learns to well on the training data but sometimes not generalizes on validation data.
3. For U-Net, the IoU gradient is steep at first but steeply decreases to lower numbers thereafter, as is typical of diminishing returns as epochs go on. The fluctuating gradient values suggest that the model's segmentation accuracy does not improve consistently and may plateau at certain points.
4. Epoch-wise IoU and PA improvements in Figure 11 provide a clear picture of how the model is progressing:

   a) The IoU improvement percentage shows a significant increase in the early epochs, with over 50% improvement in the first few training cycles. However, after the initial boost, the improvements slow down, and subsequent epochs exhibit fluctuating performance, indicating the model's difficulty in maintaining consistent progress.
   b) Similar to IoU, PA shows a rapid initial increase but later fluctuates with marginal improvements, highlighting the need for better optimization strategies to sustain learning progress over time.

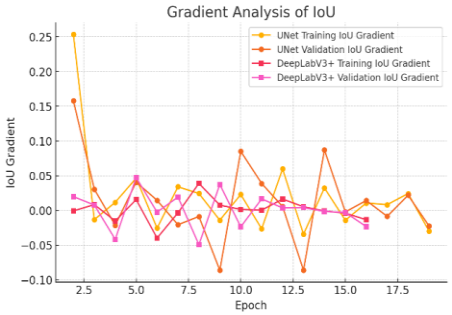


Figure 9. Gradient analysis of IoU U-Net and DeepLabV3+.

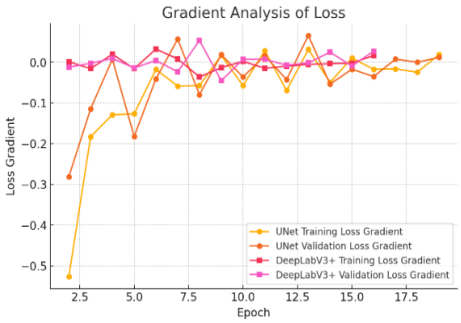


Figure 10. Gradient analysis of loss U-Net and DeepLabV3+.

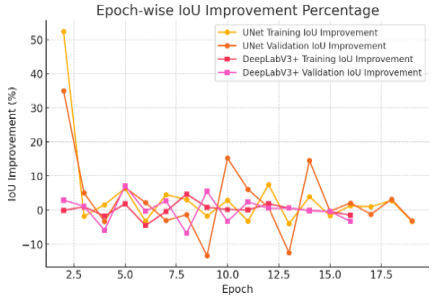


Figure 11. Epoch wise IoU improvement analysis percentage U-Net and DeepLabV3+.

4.3. DeepLabV3+ Model Performance

DeepLabV3+ is an advanced semantic segmentation model that uses Atrous convolution to capture multi-scale context information and features an improved encoder-decoder structure. The key observations from the DeepLabV3+ training process includes. DeepLabV3+ starts with a lower initial loss in Figure 3 compared to U-Net and demonstrates a more consistent decrease across epochs. This indicates efficient learning and stable optimization. The validation loss follows a relatively smooth decreasing trend, suggesting better generalization performance and lesser sensitivity to overfitting.

Both the training IoU as in Figure 6 and validation IoU as in Figure 8 indicates that they increase steadily with fewer fluctuations than U-Net, indicating a more robust segmentation performance. PA also exhibits higher values, demonstrating the model's capacity to capture finer details with precision.

The DeepLabV3+ model shows a much more stable and consistent learning behavior compared to U-Net. The main points to note are, DeepLabV3+ presents a better and more smooth decrease of the gradient, indicating a stable and consistent learning. This means that the losses have to reduce progressively, which is expected for good optimization (less abrupt changes).

Low validation loss gradient fluctuation indicates significant generalization to out-of-sample data. This lack of fluctuation indicates that the model achieves an optimal trade-off between bias and variance, minimizing the potential for overfitting. The IoU gradient is smooth, showing signs of improvement each epoch. In contrast with U-Net model, there isn't much variation in DeepLabV3+ which means improvement in segmentation over time is stable.

Figure 11 showing the epoch wise increase in IoU and PA indicating the model is learning consistently. Clear improvement of IoU (in percentage), steady and gradual increase in IoU (in percentage) during the training, and stable upward progression. It shows the model is continuing to learn and refine boundaries of segmentation with every epoch. DeepLabV3+ shows slow and steady improvements in PA, indicating that learning is effective, and improvements are realized across segmentation without suddenly declining performance.

4.4. Comparative Analysis of U-Net vs. DeepLabV3+

A comparative analysis of U-Net and DeepLabV3+ across key performance metrics offers valuable insights into their efficiency and suitability for different segmentation tasks.

Key Observations

1. The loss value of DeepLabV3+ is lower than that of U-Net, indicating earlier convergence capability. While U-Net needs more time to optimize and stabilize.
2. The IoU values for DeepLabV3+ are uniformly higher along the epochs, and candle U-Net has higher fluctuations and a slower training progress.
3. As can be seen, DeepLabV3+ achieves much higher accuracy levels and is a lot more stable, while U-Net has an inconsistent pattern.
4. While looking at the IoU data, it seems that the IoU graph for both U-Net and DeepLabV3+ models shows a good trend with respect to epochs. Observations from the graph:

   a)Since U-Net shows fairly high gradients at the beginning of the training, the IoU is improving significantly in the early epochs, therefore the learning is fast. In contrast, DeepLabV3+ enjoyed a more stable learning rate from the begin due to the smaller values of the gradients which means better stability.

   b) DeepLabV3+ exhibits more stable gradients with lesser oscillations as training progresses, indicating the model is learning at a stable pace. In contrast, U-Net shows more fluctuation, indicating that this model has not consistently learned, with sudden increases and decreases in performance.

   c)This pattern is especially evident for the validation IoU gradients of the two models, which is fluctuating for the two models but is smooth for the best-performing model (DeepLabV3+) indicating it generalizes better to unseen data.

5. The loss gradient analysis graph in Figure 10 provides insights into how quickly the training and validation loss are decreasing over epochs.

   a) U-Net starts with a steep gradient, implying that it achieves the loss quickly at the beginning of training. This indicates that the model learns the training data very quickly but also might show signs of overfitting because there is no early stopping for that.

   b) DeepLabV3+ provides more consistent and constant gradients over the course of the training period without sudden jumps in the gradients.

   c) Notice how DeepLabV3+ gets to stable validation loss gradients much sooner than U-Net, which does not seem fully converged (still variation) yet.

This graph in Figure 11 highlights the % improvement in IoU for both training and validation data across epochs.

The Table 1 indicates U-Net model exhibits a significant improvement in the first few epochs, with IoU percentage improvement reaching over 50%. However, after the initial spike, the rate of improvement slows down and fluctuates, indicating inconsistent learning behavior. DeepLabV3+ demonstrates a more gradual and consistent improvement percentage over

time, with smaller fluctuations. This suggests that the model is improving steadily without sudden changes in performance. While U-Net experiences a rapid boost initially, its performance gains become inconsistent. In contrast, DeepLabV3+ maintains a steady upward trajectory, indicating more reliable learning. These insights suggest that while U-Net can achieve rapid early-stage performance improvements, DeepLabV3+ provides a more reliable and generalizable solution over longer training periods.

Table 1. Various metrics of U-Net and deeplabV3+.

Model	U-Net	DeepLabV3+
Training loss	0.18	0.13
Validation loss	0.27	0.30
Training IoU	0.85	0.89
Validation IoU	0.69	0.69
Training PA	0.94	0.96
Validation PA	0.91	0.89
Validation dice	0.83	0.84
Training dice	0.94	0.95

Table 2. Various aspects of U-Net and DeepLabV3+ based on metrics.

Aspect	DeepLabV3+	U-Net	Improvement% (DeepLabV3+ vs U-Net)
Model stability	Stable improving consistently	Initial improvements only	4.7%
Generalization	Smooth trends better generalization	Fluctuates but good	11%
Learning efficiency	Steady improvements	Fast and only initial improvements	28%

From the Table 2, the observations are:

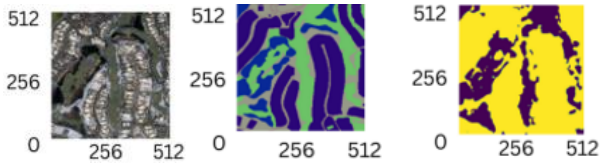
1. Based on training loss, it can be concluded that DeepLabV3+ converges faster
2. Based on validation loss, it can be concluded that DeepLabV3+ generalizes better
3. Based on training IoU, it can be concluded that DeepLabV3+ achieves better accuracy
4. Based on validation IoU, it can be concluded that DeepLabV3+ shows better consistency
5. Based on training PA, it can be concluded that DeepLabV3+ provides better learning.
6. Based on validation PA, it can be concluded that DeepLabV3+ is more reliable

The 0.89 IoU value shown by DeepLabV3+ indicates a strong capability in boundary detection, which is crucial for applications in urban planning such as zoning, road extraction, and infrastructure mapping. Similarly, a PA of 0.96 indicates that the model is capable of accurately classifying the majority of pixels, which is crucial for monitoring extensive areas. The enhanced accuracy achieved with reduced manual correction time will facilitate improved decision-making in environmental and disaster response sectors.

However, from Table 1 we can observe that, U-Net remains a viable option for simpler tasks, especially when computational resources are limited. It may require additional tuning and regularization techniques to achieve similar levels of performance compared to DeepLabV3+. Choosing between these models depends on the complexity of the segmentation task, data availability, and computational constraints.

a) Original image. b) Ground truth. c) U-Net Predicted mask.

Figure 12. Original image vs ground truth vs predicted mask using U-Net.



a) Original image. b) Ground truth. c) DeeplabV3+ Predicted mask.

Figure 13. Original image vs ground truth vs predicted mask using DeepLabV3+.

From the visual analysis shown in Figures 12 and 13, it is evident that DeepLabV3+ outperforms U-Net in terms of learning efficiency, stability, and generalization capabilities. DeepLabV3+ consistently achieves lower loss, higher IoU, and better PA with fewer fluctuations. This makes it an ideal choice for applications requiring high precision and robustness. In the analysis of DL models, visualizing performance metrics is crucial to understanding model behavior across different training and validation phases. The following set of plots (as in shown Figures15 to 18 and 20 to 23) provides insights into the performance of DenseNet201-CNN and VGG16 models applied to a multi-class classification task. These visualizations help identify trends, potential overfitting, and areas of improvement by examining various metrics such as accuracy, loss, and class-wise performance. Also, these visualizations collectively provide a comprehensive understanding of model strengths and weaknesses, facilitating informed decisions on hyperparameter tuning, data preprocessing, and potential improvements to model architecture.

4.5. DenseNet201-CNN

Dataset [18] for classification consist of total 5630 images of then size 512X512 taken from RSI-CB256 dataset. This dataset has RSIs in four classes. For training of DenseNet201-CNN model, dataset splitted into 4053 images for train, 1014 images for validation, and 564 images for test. But VGG16 model use different split with 1126 images for test and other pictures for train. This is represented in Figure 14. Such way gives good balance between training and checking both models, help learning and generalizing better.

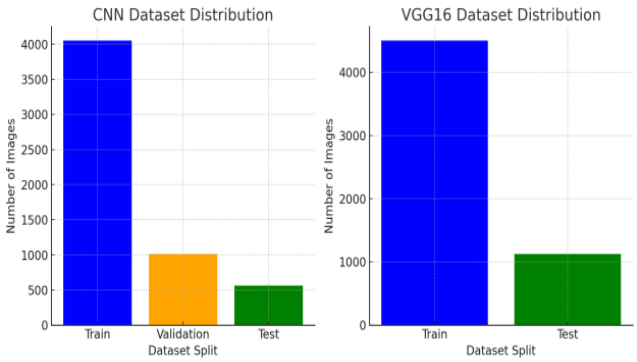


Figure 14. Dataset distribution for classification (DenseNet201-CNN and VGG16).

This plot in Figure 15 displays the training and validation accuracy across multiple epochs, providing insights into how well the model learns over time.

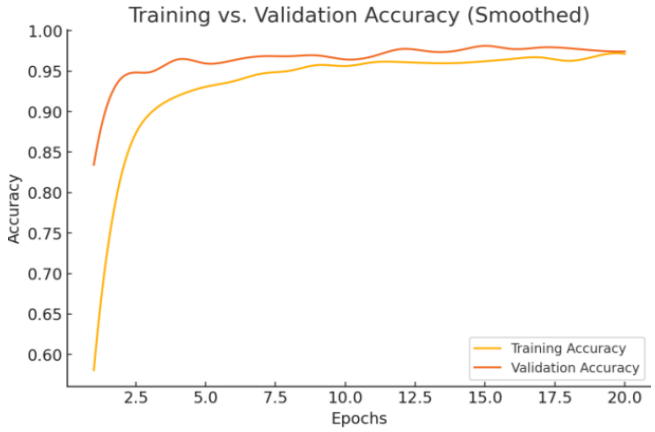


Figure 15. Accuracy over epochs of DenseNet201-CNN model.

A combined visualization of training and validation accuracy along with loss trends over epochs. This plot in Figure 16 highlights whether the model converges effectively and identifies any discrepancies between the two datasets.

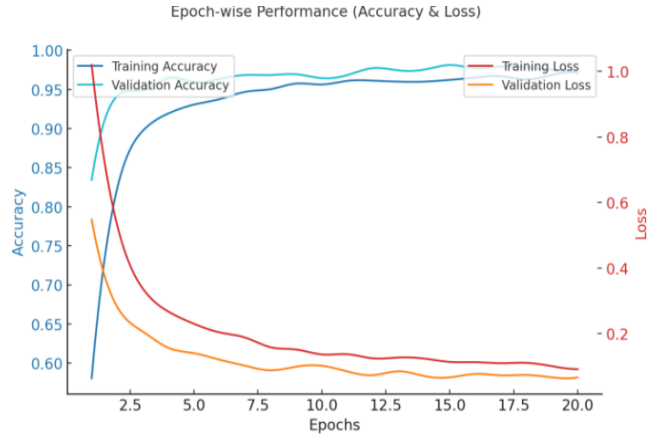


Figure 16. Epoch wise performance comparison of DenseNet201-CNN model.

A class-wise breakdown of F1, F2, and F0.5 scores, which represent the balance between precision and recall for different classes. This plot in Figure 17 helps to pinpoint which categories perform well and where misclassifications might be occurring.

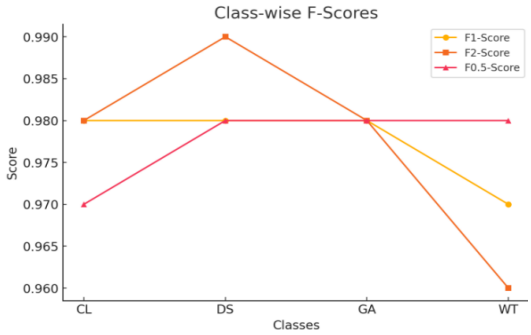


Figure 17. Class wise F-scores of DenseNet201-CNN model.

A depiction of the training and validation loss trends, showcasing how the model minimizes error over time. A consistently high validation loss compared to training loss is observed in Figure 18.

Figure 18. Loss over epochs of DenseNet201-CNN model.

The DenseNet201-CNN model's confusion matrix Figure 19 shows how good it is at classifying by looking at predicted labels and real ones. It classifies in four classes: Cloudy, desert, green area, and water. When diagonal shows up, it means right classifications happen.

1. Cloudy.

- *Correct classifications*: 147 images correctly classified as "cloudy."
- *Incorrect classifications*: 3 images incorrectly classified as "desert."

2. Desert.

- *Correct classifications*: 113 images correctly classified as "desert."
- *Incorrect classifications*: 1 image incorrectly classified as "cloudy."

3. Green_Area.

- *Correct classifications*: 148 images correctly classified as "green_area."
- *Incorrect classifications*: 2 images incorrectly classified as "water."

4. Water.

- *Correct classifications*: 143 images correctly classified as "water."

- *Incorrect classifications*

a) 3 images incorrectly classified as "cloudy."
b) 4 images incorrectly classified as "green_area."

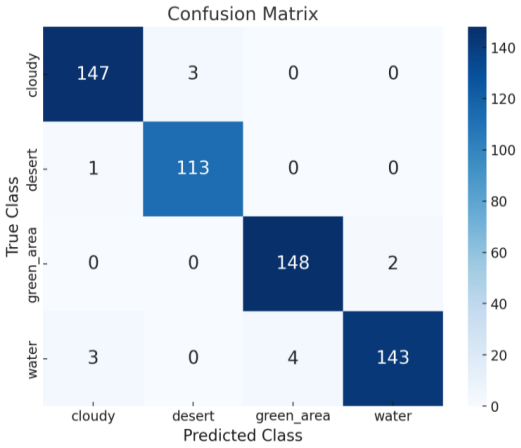


Figure 19. Confusion matrix of DenseNet201-CNN model.

The Table 3 shows how well four classes do: Cloudy (CL), Desert (DS), Green Area (GA), and Water (WT). The DenseNet201-CNN model demonstrates high performance with an overall accuracy of 97.70%, indicating its strong ability to correctly classify different environmental categories. Precision, which measures the proportion of correctly identified positive cases, is high across all classes, with values ranging from 0.973 (green_area) to 0.986 (water). A high precision means the model makes fewer false positive errors. Conversely, the False Discovery Rate (FDR), which represents the proportion of incorrect positive predictions, remains low, with the highest being 0.026 (green_area), indicating that most predictions made by the model are reliable.

Table 3. Class wise metrics of DenseNet201-CNN model.

	CL	DS	GA	WT
Precision	0.97	0.97	0.97	0.99
1-precision (FDR)	0.03	0.03	0.03	0.01
Recall (TPR)	0.98	0.99	0.99	0.95
1-recall (FNR)	0.02	0.01	0.01	0.05
F1-Score	0.98	0.98	0.98	0.97
Specificity (TNR)	0.99	0.99	0.99	1.00
Balanced accuracy	0.99	0.99	0.99	0.97
F2-score	0.98	0.99	0.98	0.96
F0.5-score	0.97	0.98	0.98	0.98
G-mean	0.98	0.99	0.99	0.97

Recall True Positive Rate (TPR), which measures how well the model identifies actual positive cases, is also notably high, with desert having the highest recall at 0.991, meaning the model correctly identified 99.1% of all actual desert cases. The False Negative Rate (FNR), which represents the proportion of missed positive cases, is consequently low, with the green_area class having an FNR of just 0.013, meaning only 1.33% of green_area instances were misclassified. The F1-score, which balances precision and recall, remains consistently high, ensuring that the model performs well in both aspects.

Furthermore, the Specificity True Negative Rate (TNR) is quite strong, as evidenced by the fact that the model is consistently identifying negative cases for all classes. The water class has the highest specificity at 0.995, which means there are very few other classes misclassifying them as water. Balanced accuracy, the mean of recall and specificity, and which does not get overly affected by the presence of classes with few samples, sits above 0.97 across all classes. F2-score that weighs recall greater than precision, and F0. 5-score, a new performance metric that weights towards precision, indicate the model preserves robust predictive vigor, irrespective of recall vs. precision weigh in.

The Geometric Mean (G-Mean), balances between sensitivity (recall) and specificity adds extra evidence of the model performance, approaching 0.99 for most classes. In summary, the model achieves an outstanding trade-off between precision, recall and specificity, so false positives and negatives are kept to a minimum. The desert achieves the best classification performance, while the water class has strong precision but slightly lower recall (0.953), meaning that some water pixels are likely misclassified. Nonetheless, the high balanced accuracy, and F1-scores in all classes confirm the model's high effectiveness for environmental classification tasks.

4.5.1. Cross-Validation Performance

To further evaluate the generalization ability of the DenseNet201-CNN model and mitigate concerns of potential overfitting, a 5-fold stratified cross-validation was conducted. This approach maintains balanced class distribution across each fold, enabling a comprehensive evaluation over diverse subsets of the dataset. The model achieved accuracies of 0.9852, 0.9864, 0.9852, 0.9877, and 0.9901 across the five folds, respectively. The overall mean accuracy was 0.9869, with a standard deviation of just ±0.0018, indicating exceptional consistency. These results confirm the model's robustness and its ability to generalize well across unseen data, thereby directly addressing earlier concerns of overfitting that were implied by the single-split test accuracy of 97.69%.

4.6. VGG16

The five key plots presented include: the plot of "accuracy over epochs" in Figure 20 shows how the model accuracy evolved on training and validation over five epochs. At first, the training accuracy is low but continues to rise until somewhere around epoch four when it plateaus indicating that the model is learning the underlying structure of the data. The validation accuracy is not smooth in early epochs, instead it keeps changing and hovering, which suggests inconsistency in learning at this stage, could be due to data variability.

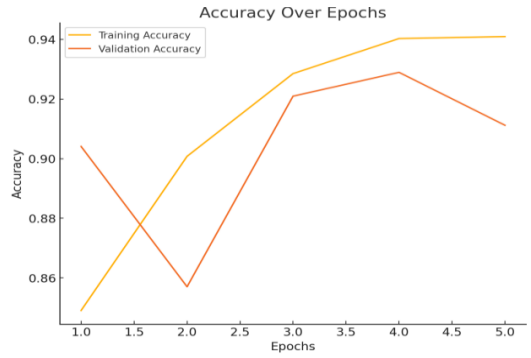


Figure 20. Accuracy over epochs of VGG16 model.

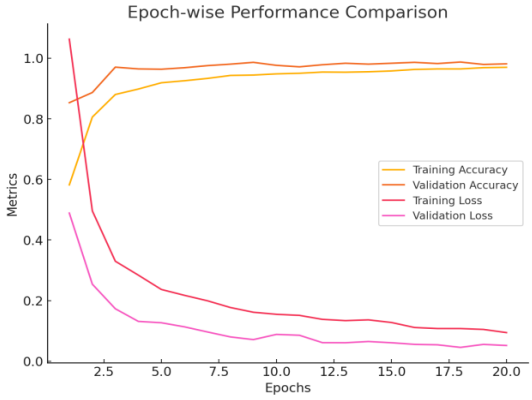


Figure 21. Epoch wise performance of VGG16 model.

As can be seen from the "epoch-wise performance comparison" in Figure 21, the accuracy and loss values for the training and validation sets for each epoch. It can be seen that the training and validation accuracies are both going up continuously, and the validation accuracies are closely following the training accuracies, meaning the model is able to learn appropriately.

Figure 22. Class wise F-Scores of VGG16 model.

The "F-scores by class" plot shows how many F-scores (F1, F2, and F0.5) were calculated to find the best balance between precision and recall for each prediction. 5). In the results presented in Figure 22, the model demonstrates improved performance on the "green_area" class with the highest F-scores which indicates a balanced between precision and recall. In contrast the "desert" class has lowest scores suggesting the difficulty to make accurate predictions on desert samples, probably because its features overlap with other classes like "cloudy" or "green_area." F-score discrepancies signify divergent behaviors in class-specific strengths and weaknesses, likely due to skewed class distributions or feature sets across categories.

The "loss over epochs" plot in Figure 23 illustrates how the model's error decreases over time for both the training and validation datasets. The training loss follows a downward trajectory, indicating successful learning and optimization of the model's weights.

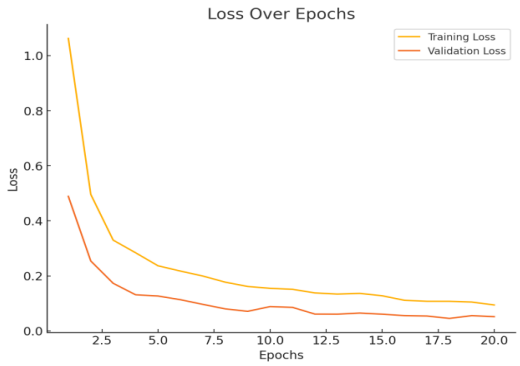


Figure 23. Loss over epochs of VGG16 model.

- **VGG16**: the confusion matrix of the VGG16 model as shown in Figure 24. The confusion matrix of the VGG16 model illustrates its classification performance by comparing predicted and actual labels. It falls into four categories: cloudy, desert, green_area, and water. The diagonal numbers indicate the correct classifications.

1. Cloudy.

- *Correct classifications*: 241 images correctly classified as "cloudy."
- *Incorrect classifications*

a) 56 images incorrectly classified as "desert."
b) 1 image incorrectly classified as "green_area."
c) 2 images incorrectly classified as "water."

2. Desert.

- *Correct classifications*: 226 images correctly classified as "desert."
- *Incorrect classifications*

a)There is no incorrect classification

3. Green_Area.

- *Correct classifications*: 283 images correctly classified as "green_area."
- *Incorrect classifications*

a) 1 image incorrectly classified as "desert."
b) 16 images incorrectly classified as "water."

4. Water.

- *Correct classifications*: 270 images correctly classified as "water"
- *Incorrect classifications*

a) 11 images incorrectly classified as "cloudy."
b) 19 images incorrectly classified as "green_area."

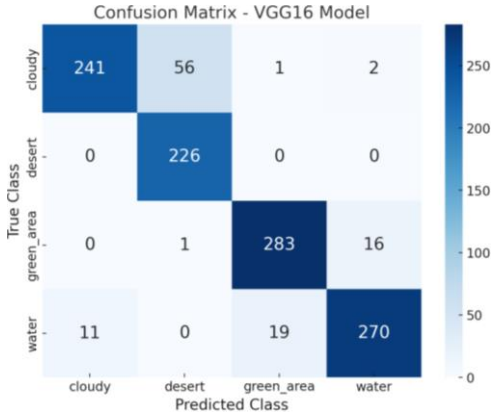


Figure 24. Confusion matrix of VGG16 model.

The Table 4 shows how well four different classes did: CL, DS, GA, and WT. The things looked at are precision, recall, F1-score, specificity, and some other measures. The "desert" class has perfect precision of 1.00, meaning every guess for this class was right. But the "cloudy" class has a lower score of 0.80 in precision; this means more wrong guesses where non-cloudy images were wrongly called cloudy. This is shown by the high FDR for "cloudy," which is at 0.20 or 20%, showing one in five predictions was wrong.

Table 4. Class wise metrics of VGG16 model.

	CL	DS	GA	WT
Precision	0.80	1.00	0.94	0.90
1-Precision (FDR)	0.20	0.00	0.06	0.10
Recall (TPR)	0.96	0.80	0.93	0.94
1-Recall (FNR)	0.04	0.20	0.07	0.06
F1-score	0.87	0.89	0.94	0.92
Specificity (TNR)	0.80	1.00	0.94	0.90
Balanced accuracy	0.88	0.90	0.94	0.92
F2-score	0.92	0.83	0.94	0.93
F0.5-score	0.83	0.95	0.94	0.91
G-mean	0.88	0.89	0.94	0.92

For finding actual positive cases correctly, the "cloudy" class shines with a score of 0.96, proving it does well in spotting most real cloudy situations. Yet the recall for the "desert" class drops to 0.80; thus, they missed about 20% of actual desert instances as per its FNR standing at 0.20.

The highest F1-score belongs to "green area" with a value of 0.94 that shows it balanced between precision and recall nicely for this group of data points. Meanwhile, "cloudy" takes home the lowest balanced accuracy at only 0.88 and that's mainly due to not-so-great specificity.

When looking into F2-score where remembering true positives matters more it peaks for both "cloudy" and "water" with scores reaching around then making a case that these groups get found better without many false negatives showing up each time too much misclassification occurs either way here respectively overall too badly still there needs some fine-tuning done ahead still though likely particularly concerning balancing them out later on again overall across perhaps easier ways additionally also move forward coming back together soon hence wise

Lastly reflecting G-mean values gauging blend touching sensibility and specificity across variations those output ratings align closely hovering between digits. The G-mean of the model is very good for "green area" and "water" classes. This performance can be viewed as comparable with the study [2] that applies fuzzy C-means clustering on CIE L*a*b*-based segmentation, integrating Landsat and Google Earth imagery, to analyze long-term land cover changes in Saudi Arabia (1984-2018).

4.7. Dataset Dynamics

Despite the high accuracy produced by segmentation and classification models, the RSI is affected by dynamic environmental conditions, including seasonal changes, cloud coverage, and illumination differences. Noise may introduce such variations, which could impact the model's generalization in real-world operations. For future work, temporal augmentation or ensemble learning could effectively address these variations.

5. Conclusions

DL is a powerful approach suited to the properties of RSI that can be used for segmentation and classification tasks. With many high-resolution RSIs, better models are needed to understand complicated scenes correctly. This study focuses on to evaluating the performance of types of DL such as U-Net and DeepLabV3+ for segmentation, DenseNet201-CNN, VGG16 for the classification of images. The goal is to evaluate the ability of these models to handle challenging RS data problems, including texture variability, size variation, and varying environmental conditions. This study provides some insights into what model best suits a myriad of RS applications.

Compared to U-Net, DeepLabV3+ performed better in segmentation, achieving higher IoU and PA scores, as well as more stable learning. U-Net initially performed good, but it had variable performance meaning that it could be sensitive to hyperparameter configurations and also overfit. Because DeepLabV3+ is available on new information it can be a more suitable choice for a job that always wants very total and strong segmentation. From the gradient study, DeepLabV3+ seems to learn smoothly (no issues) and U-Net needs additional regularization and augmentations to be unshakable (regularization and set augmentation).

DenseNet201-CNN outperformed VGG16 in classification accuracy, a clear representation of meaningful complex space features and patterns. The misclassification rates were lower with DenseNet201-CNN compared with VGG16, indicating its superior feature extraction capabilities. Although fine for simple

tasks, VGG16 lacks the strategic depth and (subsequent) efficient feature extraction leveraged in DenseNet201-CNNs making it not the best model for challenging datasets.

As far as choosing model, it has to consider computation efficiency and how accurate results should be, for image segmentation and classification purposes, DeepLabV3+ and DenseNet201-CNN are two great choices.

Ethical Compliance

This manuscript is an original work that has not been published previously nor is under consideration elsewhere. While it references prior studies, it differs in model configurations, dataset compositions, and experimental protocols, providing a novel comparative evaluation using DeepLabV3+, U-Net, DenseNet201-CNN, and VGG16 for RS segmentation and classification tasks.

References

[1] Ali I., Rehman A., Khan D., Khan Z., and et al., "Model Selection Using K-Means Clustering Algorithm for the Symmetrical Segmentation of Remote Sensing Datasets," *Symmetry*, vol. 14, no. 6, pp. 1-19, 2022. https://doi.org/10.3390/sym14061149

[2] Alzahrani A. and Bhuiyan A., "On Satellite Imagery of Land Cover Classification for Agricultural Development," *The International Arab Journal of Information Technology*, vol. 20, no. 1, pp. 9-18, 2023. https://doi.org/10.34028/iajit/20/1/2

[3] Bilgin G., Erturk S., and Yildirim T., "Segmentation of Hyperspectral Images via Subtractive Clustering and Cluster Validation Using One-Class Support Vector Machines," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 8, pp. 2936-2944, 2011. DOI: 10.1109/TGRS.2011.2113186

[4] Chen L., Zhu Y., Papandreou G., Schroff F., and Adam H., "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation," *in Proceedings of the 15th European Conference on Computer Vision*, Munich, pp. 833-851, 2018. https://doi.org/10.1007/978-3-030-01234-2_49

[5] Cheng G., Han J., and Lu X., "Remote Sensing Image Scene Classification: Benchmark and State of the Art," *Proceedings of the IEEE*, vol. 105, no. 10, pp. 1865-1883, 2017. DOI: 10.1109/JPROC.2017.2675998

[6] Cheng G., Xie X., Han J., Guo L., and Xia G., "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

[7] Guo M., Xu T., Liu J., Liu Z., and et al., "Attention Mechanisms in Computer Vision: A Survey," *Computational Visual Media*, vol. 8, no. 3, pp. 331-368, 2022. https://doi.org/10.1007/s41095-022-0271-y

[8] Hong D., Gao L., Hang R., Zhang B., and Chanussot J., "Deep Encoder-Decoder Networks for Classification of Hyperspectral and LiDAR Data," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1-5, 2022. DOI: 10.1109/LGRS.2020.3017414

[9] Humans in the Loop, Kaggle, Semantic Segmentation of Aerial Imagery, https://www.kaggle.com/datasets/humansintheloop/semantic-segmentation-of-aerial-imagery, Last Visited, 2025.

[10] Jiang Y., Liu S., and Wang H., "Diffusion-based Remote Sensing Image Fusion for Classification," *Applied Intelligence*, vol. 55, no. 4, pp. 247, 2025. https://doi.org/10.1007/s10489-024-06217-z

[11] Lowe D., "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91-110, 2004. https://doi.org/10.1023/B:VISI.0000029664.99615.94

[12] Maldonado S., Carrizosa E., and Weber R., "Kernel Penalized K-Means: A Feature Selection Method Based on Kernel K-Means," *Information Sciences*, vol. 322, pp. 150-160, 2015. https://doi.org/10.1016/j.ins.2015.06.008

[13] Michael S. and Wu Y., "Remote Sensing Image Classification with the SEN12MS Dataset," *arXiv Preprint*, vol. arXiv:2104.00704v1, pp. 1-6, 2021. https://arxiv.org/abs/2104.00704

[14] Niazmardi S., Demir B., Bruzzone L., Safari A., and Homayouni S., "Multiple Kernel Learning for Remote Sensing Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 3, pp. 1425-1443, 2018. DOI: 10.1109/TGRS.2017.2762597

[15] Pan J., Wei Z., Zhao Y., Zhou Y., and et al., "Enhanced FCN for Farmland Extraction from Remote Sensing Image," *Multimedia Tools and Applications*, vol. 81, pp. 38123-38150, 2022. https://doi.org/10.1007/s11042-022-12141-6

[16] Pham L., Tran K., Ngo D., Lampert J., and Schindle A., "Remote Sensing Image Classification using Transfer Learning and Attention Based Deep Neural Network," *arXiv Preprint*, vol. arXiv:2206.13392v1, pp. 1-6, 2022. https://doi.org/10.48550/arXiv.2206.13392

[17] Rasti B., Ghamisi P., and Ulfarsson M., "Hyperspectral Feature Extraction Using Sparse and Smooth Low-Rank Analysis," *Remote*

Sensing, vol. 11, no. 2, pp. 1-21, 2019. https://doi.org/10.3390/rs11020121

[18] Reda M., Satellite Image Classification, Kaggle, https://www.kaggle.com/datasets/mahmoudreda55/satellite-image-classification, Last Visited, 2025.

[19] Ren Y., Li X., Yang X., and Xu H., "Development of a Dual-Attention U-Net Model for Sea Ice and Open Water Classification on SAR Images," *IEEE Geoscience and Remote Sensing Letters*, vol 19, pp. 1-5, 2022. DOI: 10.1109/LGRS.2021.3058049

[20] Ronneberger O., Fischer P., and Brox T., "U-Net: Convolutional Networks for Biomedical Image Segmentation," *in Proceedings of the 18th International Conference on Medical Image Computing and Computer-Assisted Intervention*, Munich, pp. 234-241, 2015. https://doi.org/10.1007/978-3-319-24574-4_28

[21] Tao C., Qi J., Li Y., Wang H., and Li H., "Spatial Information Inference Net: Road Extraction Using Road-Specific Contextual Information," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 158, pp. 155-166, 2019. https://doi.org/10.1016/j.isprsjprs.2019.10.001

[22] Thapa A., Horanont T., Neupane B., and Aryal J., "Deep Learning for Remote Sensing Image Scene Classification: A Review and Meta-Analysis," *Remote Sensing*, vol. 15, no. 19, pp. 1-37, 2023. https://doi.org/10.3390/rs15194804

[23] Tombe R. and Viriri S., "Remote Sensing Image Scene Classification: Advances and Open Challenges," *Geomatics*, vol. 3, pp. 137-155, 2023. https://doi.org/10.3390/geomatics3010007

[24] Villa A., Benediktsson J., Chanussot J., and Jutten C., "Hyperspectral Image Classification with Independent Component Discriminant Analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 12, pp. 4865-4876, 2011. DOI: 10.1109/TGRS.2011.2153861

[25] Wang X., Zhu J., Yan Z., and Zhang Z., "LaST: Label-Free Self-Distillation Contrastive Learning with Transformer Architecture for Remote Sensing Image Scene Classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1-5, 2022. DOI: 10.1109/LGRS.2022.3185088

[26] Xia G., Hu J., Hu F., Shi B., and et al., "AID: A Benchmark Data Set for Performance Evaluation of Aerial Scene Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3965-3981, 2017. DOI: 10.1109/TGRS.2017.2685945

[27] Xu K., Deng P., and Huang H., "Vision Transformer: An Excellent Teacher for Guiding Small Networks in Remote Sensing Image Scene Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1-15, 2022. DOI: 10.1109/TGRS.2022.3152566

[28] Xu Z., Zhang W., Zhang T., Yang Z., and Li J., "Efficient Transformer for Remote Sensing Image Segmentation," *Remote Sensing*, vol. 13, no. 18, pp. 1-24, 2021. https://doi.org/10.3390/rs13183585

[29] Yang Z., Wu Q., Zhang F., Zhang X., and et al., "A New Semantic Segmentation Method for Remote Sensing Images Integrating Coordinate Attention and SPD-Conv," *Symmetry*, vol. 15, no. 5, pp. 1-17, 2023. https://doi.org/10.3390/sym15051037

[30] Zhang J., Lin S., Ding L., and Bruzzone L., "Multi-Scale Context Aggregation for Semantic Segmentation of Remote Sensing Images," *Remote Sensing*, vol. 12, no. 4, pp. 1-16, 2020. https://doi.org/10.3390/rs12040701

[31] Zhang Y., Bai X., Fan R., and Wang Z., "Deviation-Sparse Fuzzy C-Means with Neighbor Information Constraint," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 1, pp. 185-199, 2019. DOI: 10.1109/TFUZZ.2018.2883033

[32] Zheng X. and Chen T., "High Spatial Resolution Remote Sensing Image Segmentation Based on the Multiclassification Model and the Binary Classification Model," *Neural Computing and Applications*, vol. 35, pp. 3597-3604, 2023. https://link.springer.com/article/10.1007/s00521-020-05561-8

Saivenkatalakshmi Ananth is a Research Scholar pursuing PhD in the area of Artificial Intelligence and Machine Learning for Remote Sensing Data Analysis from Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Hyderabad-500075, Telangana, India.

Suryakanth Gangashetty is a Professor in Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India Completed PhD from IIT Madras and post-doc from CMU, Pittsburgh, USA. Interests are in areas of Artificial Intelligence, Machine Learning, Human Computer Interaction, Speech Signal Processing.