**JOIN (Jurnal Online Informatika)**
p-ISSN: 2528-1682, e-ISSN: 2527-9165
Volume 10 Number 2 | December 2025: 286-298
DOI: 10.15575/join.v10i2.1612

# Land Cover Classification in Mountainous Regions Using Multi-Scale Fusion and Convolutional Neural Networks: A Case Study on Mount Slamet

**Yulis Rijal Fauzan[1], Shofwatul 'Uyun[2]**
[1,2] Master of Informatics, Faculty of Science and Technology, UIN Sunan Kalijaga, Yogyakarta, Indonesia

**Article Info**

***Article history:***

Received April 17, 2025
Revised May 21, 2025
Accepted May 31, 2025
Published August 17, 2025

***Keywords:***

CNN
DenseNet121
Guided Filter
Land Cover Classification
MobileNetV2
Multi-Scale Fusion
VGG-16

**ABSTRACT**

Mount Slamet, located in Central Java, Indonesia, is a high-risk volcanic region where accurate land cover classification is essential for disaster mitigation and sustainable land management. However, satellite imagery in this area often suffers from haze and cloud cover, posing challenges to reliable classification. This study aims to develop an effective land cover classification model using Sentinel-2 imagery by addressing these visual distortions. The specific goal is to classify land cover into five classes—Forest, Settlements, Summit, RiceField, and River—using enhanced satellite images. A total of 1101 labeled images were processed through dehazing with Multi-Scale Fusion (MSF) and smoothing using a Guided Filter to improve image quality. The classification was performed using three Convolutional Neural Network (CNN) architectures: VGG-16, MobileNetV2, and DenseNet121. The main contribution of this study is the integration of a tailored preprocessing pipeline with CNN-based modeling for haze-affected mountainous satellite imagery. Among the models tested, MobileNetV2 achieved the highest accuracy of 85.4%, outperforming DenseNet121 (83.8%) and VGG-16 (82.3%). The results demonstrate the effectiveness of combining image enhancement techniques with lightweight CNN architectures for land cover classification in challenging environments with limited and imbalanced dataset.

***Corresponding Author:***

Yulis Rijal Fauzan,
Master of Informatics, Faculty of Science and Technology, UIN Sunan Kalijaga, Yogyakarta, Indonesia
Jl. Laksda Adisucipto, Papringan, Caturtunggal, Daerah Istimewa Yogyakarta 55281
Email: yulisrf41@gmail.com

## 1. INTRODUCTION

Mount Slamet is one of a group of mountains located in western Indonesia, in a series of mountains on the island of Java [1]. Mount Slamet reaches an altitude of 3,432 meters above sea level and is located within the administrative regions of Pemalang, Brebes, Tegal, Banyumas, and Purbalingga. Mount Slamet has weak explosive and effusive eruptions, which tend to be less hazardous to the surrounding agricultural areas and settlements [2]. This has led to the conversion of land around the mountain into agricultural and settlement areas to meet the social and economic needs of the local population [3]. Reduction in land area and changes in its function can result in decreased water absorption capacity, increased erosion risk, and soil fertility degradation [4]. Mount Slamet has a forest area of 52,617 hectares, with the largest area in Banyumas Regency reaching 9,887.6 hectares. Over 11 period from 2008years19, there was a reduction in forest land of approximately 913.96 hectares, or about 9.24% of the total forest area [5]. This change requires accurate monitoring to understand its impact on the ecosystem and land use.

To understand the dynamics of land cover changes, accurate analysis through satellite image classification is needed. This technique allows for the identification and mapping of land cover based on spectral characteristics [6]. Satellite image classification is a process of grouping data into specific categories based on certain characteristics or features. This method can be used to map land cover or environmental changes based on the spectral values of satellite image pixels, using machine learning or deep learning algorithms [7][8].

Deep learning is an advancement from Artificial Neural Networks (ANN) with deeper and more complex networks. Convolutional Neural Network (CNN) is one of the deep learning algorithms commonly used for image analysis due to its ability to recognize patterns or objects from large image data sets [9][10]. This process involves breaking down the image into basic features like edges and blobs at the initial layers, and then representing more complex patterns in subsequent layers [11].

This study aims to classify land cover in the Mount Slamet region, specifically Banyumas Regency, using Sentinel-2 satellite imagery. The research dataset consists of 1101 labeled data divided into five data classes. Before classification, preprocessing was performed using Multi-Scale Fusion (MSF) to remove haze effects (dehazing) from the images, resulting in clearer images while preserving the original colors [12], and Guided Filter, which is effective for noise removal, enhancing contrast, and preserving edge details (smoothing) (He et al., 2013). Additionally, to address the challenge of data imbalance, data augmentation was applied to expand the training data variation, thereby improving the performance and generalization of the model [13].

Classification was carried out using CNN architectures, such as VGG-16, MobileNetV2, and DenseNet, each selected for their ability to handle relatively small datasets. VGG-16 allows for complex feature extraction through strong transfer learning [14]. MobileNetV2 features an efficient architecture that adapts well to limited datasets [15]. DenseNet excels in optimizing feature information through dense connections between layers, significantly improving performance on smaller datasets [16].

To support this research, a literature review is provided to discuss various satellite image classification approaches and implementations of CNN architectures. Previous studies have shown the effectiveness of CNNs in various land cover classification applications, but their application to the local challenges of Mount Slamet, such as haze and clouds common in tropical mountain areas, has been less explored.

Research conducted by Patrick Helber et al. used CNN algorithms with ResNet-50 architecture on the EuroSAT dataset, achieving an accuracy of 98.57% for land cover classification [17]. Geetha M. et al. implemented CNN architectures, including VGG16, ResNet34, and a custom model on Sentinel-2 satellite imagery, with the custom model achieving the highest accuracy of 93.73% [18]. Mandicou B.A. et al. developed CNN models based on U-Net and FCN8 architecture for satellite image classification in the Senegal River Valley, with U-Net reaching the highest accuracy of 96.10% [19]. Sana Basheer et al. evaluated SVM and Maximum Likelihood algorithms on PlanetScope Super Dove sensor (PSB.SD) dataset, achieving the highest accuracy of 94.00% with SVM [20]. Smita Sunil Burrewar et al. used RCNN architecture on Sentinel-2 satellite imagery in the Nagpur region, India, achieving a very high accuracy of 98.86% [21].

Although various previous studies have demonstrated the success of satellite image classification using specific algorithms, the application of CNNs to satellite imagery with local characteristics—such as frequent haze, clouds, and terrain variability in the Mount Slamet region—remains limited. This study contributes by integrating a specialized preprocessing pipeline (dehazing with Multi-Scale Fusion and smoothing with Guided Filter) with three CNN architectures (VGG-16, MobileNetV2, and DenseNet) to classify land cover in a fog-prone mountainous region using a small and imbalanced dataset. The contribution lies in addressing both visual noise and dataset limitations to improve classification performance in real-world tropical highland conditions.

## 2. METHOD

This study proposes an integrated framework for land cover classification using Sentinel-2 satellite imagery, focusing on the Mount Slamet area in Banyumas Regency. The main contributions of this study lie in the preprocessing pipeline using Multi-Scale Fusion (MSF) and Guided Filter to reduce haze and noise, as well as the evaluation of multiple CNN architectures using a balanced training and validation scheme. All experiments were conducted in (IDE) Google Colab using Python and TensorFlow. Figure 1 illustrates the overall workflow of the proposed method.

p-ISSN: 2528-1682
e-ISSN: 2527-9165

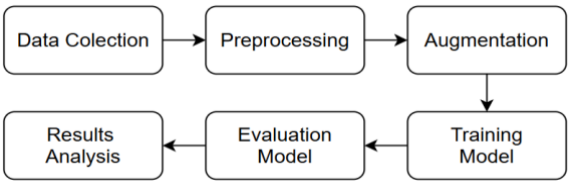


Figure 1. Research Workflow Diagram

### *2.1. Data Collection*

The dataset was collected using the Google Earth Engine (GEE) platform by extracting Sentinel-2 Level-1C imagery captured on May 5, 2024, over the Mount Slamet region in Banyumas Regency. The image acquisition process is illustrated in Figure 2.


Figure 2. Image Acquisition Using Google Earth Engine

Sentinel-2 provides 13 spectral bands with varying spatial resolutions (10 m, 20 m, and 60 m) [22]. In this study, we selected three bands corresponding to the visible RGB channels: Band 4 (Red), Band 3 (Green), and Band 2 (Blue), each with a spatial resolution of 10 meters. These bands were chosen for their effectiveness in visual interpretation and compatibility with common CNN input requirements. The visualization of the RGB channels is shown in Figure 3.

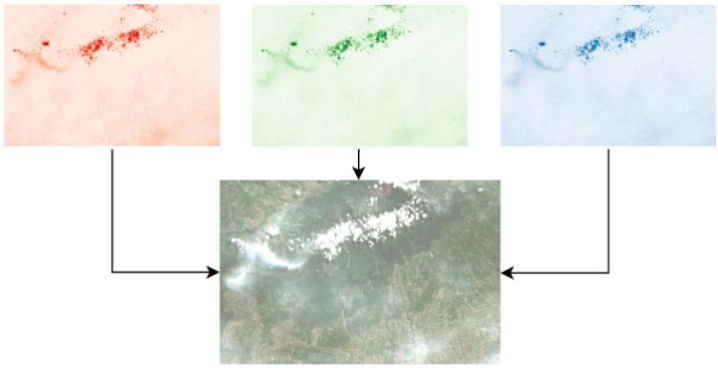

Figure 3. Visualization of the Combination of 3 RGB Channels

Next, the dataset is divided into five data classes: Forest, Settlements, Summit, RiceField, and River, with a total of 1101 data points. The distribution of data for each class can be seen in Table 1. A sample of the dataset that has been organized can be seen in Figure 4.

Table 1. Data Distribution per Class

| Class | Number of Samples |
|---|---|
| Forest | 380 |
| Settlements | 390 |
| Summit | 31 |
| RiceField | 273 |
| River | 27 |

Figure 4. Sample Dataset

### *2.2. Preprocessing*

The preprocessing stage aims to enhance the quality of satellite images by reducing visual disturbances such as haze, clouds, and noise, making the data cleaner and ready for land cover classification [23][24]. Multi-Scale Fusion (MSF) and Guided Filter are employed in this process.

#### 2.2.1. Multi-Scale Fusion (MSF)

Multi-Scale Fusion (MSF) is used to eliminate haze effects (dehazing) to produce clearer images [12]. This method employs a single-scale approach with intensity control parameters and minimum transmission thresholds to generate haze-free images. The process preserves details and natural color quality through normalization and intensity adjustment [25]. The Multi-Scale Fusion equation is presented in Equation 1.

$$J_c(x) = \frac{I_c(x) - A \cdot (1 - t(x))}{t(x)} + A \cdot (1 - t(x)) \quad (1)$$

Where:

$J_c(x)$ : Haze-free image intensity at channel c
$I_c(x)$ : Original image intensity at channel c
$A$ : Atmospheric intensity
$x$ : Transmission at pixel $x$

Multi-Scale Fusion (MSF) was chosen for its advantages in handling images with thin to moderate haze [12]. The strength of MSF lies in its ability to preserve the natural color and texture details of the image effectively. This is achieved through normalization and intensity adjustment processes, ensuring that areas with thin to moderate haze can be processed optimally without producing artifacts that compromise visual quality [26].

#### 2.2.2. Guided Filter

Guided Filter is an effective image enhancement technique for removing noise, improving contrast, and preserving edge details in the image [27]. The equation for the Guided Filter can be found in Equation 2.

$$q(x) = a(x) \cdot I(x) + b(x) \quad (2)$$

Where:

$q(x)$ : Filtered image at pixel $x$

$a(x)$ : Linear slope coefficient after smoothing
$b(x)$ : Linear bias after smoothing
$I(x)$ : Guiding image at pixel $x$

### *2.3. Augmentation*

Data augmentation is a technique used to increase the quantity and diversity of data in a dataset. It involves generating modifications of existing data, such as rotation, flipping, zooming, translation, scaling, and more [28]. In image classification, data augmentation is crucial for improving model performance, especially when training data is limited. For instance, horizontal flipping and rotation can help the model recognize objects from different perspectives [29]. Additionally, augmentation is also beneficial for addressing overfitting, a condition where the model fits the training data too closely but performs poorly on test data [30].

To address class imbalance and improve generalization, we applied augmentation techniques selectively based on the scarcity of samples per class. Augmentation is applied to the Summit and River classes, which have the fewest data points. The augmentation process includes mirror_horizontal, mirror_vertical, and rotated_180°.

### *2.4. Training Model*

The model was trained using three Convolutional Neural Network (CNN) architectures: VGG-16, MobileNetV2, and DenseNet121. Each architecture was utilized through transfer learning by loading pre-trained weights from ImageNet.

#### *2.4.1. VGG-16 Architecture*

VGG-16 is a Convolutional Neural Network (CNN) architecture consisting of 16 deep convolutional layers trained with pre-trained weights from ImageNet [31]. This architecture uses filters of the same size (3x3) that are effective for extracting basic image features, and small stride and overlap patterns to maintain spatial accuracy [32]. The VGG-16 architecture is shown in Figure 5.

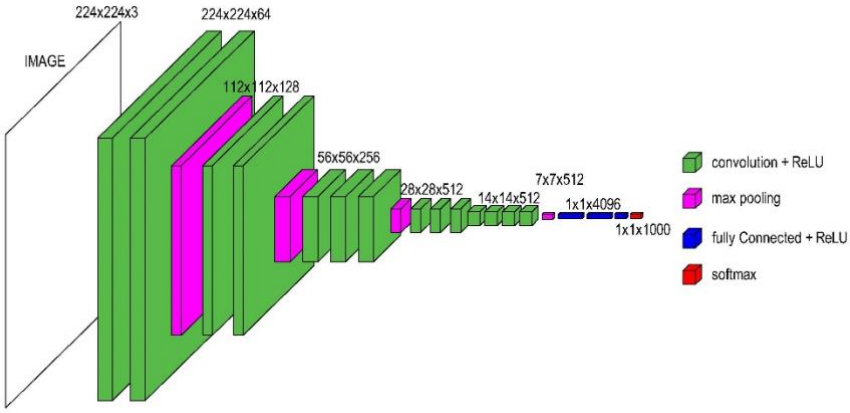


Figure 5. VGG-16 Architecture

The VGG-16 architecture consists of a total of 13 convolutional layers and 3 fully connected layers, organized in a deep and uniform structure. The original input to the VGG-16 model is a 224 × 224 × 3 RGB image, but in this study, input images were resized to 128 × 128 × 3 to accommodate computational constraints. The architecture is designed to extract hierarchical features from images through successive convolutional and pooling layers.

Each convolutional layer applies a set of filters (with increasing depth in deeper layers), followed by ReLU (Rectified Linear Unit) activation functions to introduce non-linearity. These layers are grouped into five blocks, with each block followed by a max pooling operation that reduces the

spatial dimensions while retaining the most important features. As the image propagates through the network, the spatial resolution decreases from 128 × 128 to smaller dimensions, while the depth (number of feature maps) increases.

After the convolutional blocks, the model flattens the output feature maps and passes them through a fully connected (Dense) layer with 256 neurons using ReLU activation. This is followed by a final Dense layer with a softmax activation function, whose output dimension matches the number of land cover classes (5 in this study), enabling multiclass classification.

In terms of implementation, this study employed transfer learning by loading the VGG-16 base model with pre-trained ImageNet weights while excluding the top classification layers (include_top=False). All layers in the convolutional base were frozen (layer.trainable = False) to retain previously learned features and avoid retraining them, which helps prevent overfitting especially with relatively small datasets. Only the newly added top layers were trained.

The model was compiled using the Adam optimizer with a learning rate of 1e-4, and categorical cross-entropy was used as the loss function, since the problem is a multi-class classification. To enhance generalization and prevent overfitting, early stopping was applied with a patience of 10 epochs, monitoring the validation loss. The model was trained for a maximum of 100 epochs, with the best-performing weights on the validation set being restored automatically.

Figure 5 provides a visual summary of the standard VGG-16 architecture, illustrating how spatial dimensions are progressively reduced while feature depth increases, and how the fully connected layers aggregate these features to produce class predictions.

### *2.4.2. MobileNetV2 Architecture*

MobileNetV2 is a CNN architecture specifically designed for mobile devices, offering a balance between accuracy and efficiency using the Inverted Residual block [15]. Each block expands the input image with larger convolutions and then compresses it before outputting, reducing parameters without sacrificing the quality of the recognized features [33]. The MobileNetV2 architecture can be seen in Figure 6.

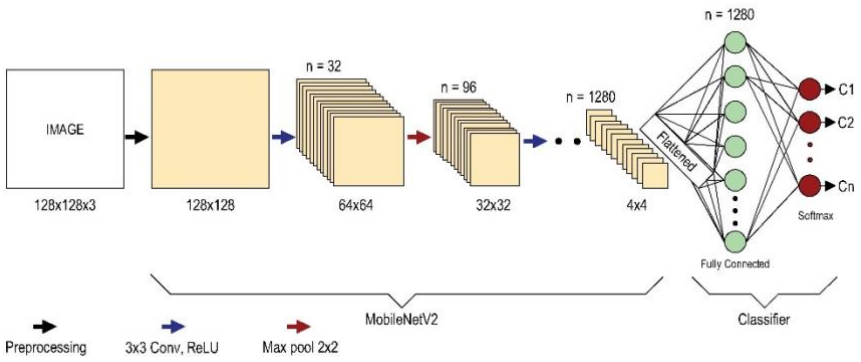


Figure 6. MobileNetV2 Architecture

The MobileNetV2 architecture is a lightweight and efficient convolutional neural network designed for fast inference on resource-constrained environments. It is structured with an emphasis on computational efficiency while maintaining performance. In this study, input images of size 128 × 128 × 3 were used, and MobileNetV2 was employed with pre-trained ImageNet weights while excluding the classification head (include_top=False).

The architecture begins with an initial 3×3 convolutional layer with a stride 2, followed by a series of depthwise separable convolution blocks, which separate spatial and channel-wise processing. Each block applies a 3×3 depthwise convolution followed by a 1×1 pointwise convolution, significantly reducing computational cost.

Central to MobileNetV2 is the use of inverted residual blocks, which expand the feature dimensions internally and then project them back to a lower dimension using linear layers. These blocks include skip connections that add the input to the output of the block in certain configurations, allowing for better gradient flow during training.

p-ISSN: 2528-1682
e-ISSN: 2527-9165

Following the convolutional backbone, the final feature map (typically of size 4×4×1280) is compressed using a global average pooling operation. In this implementation, however, the output is flattened and passed through a Dense layer with 256 neurons using ReLU activation, followed by a Dropout layer with a rate of 0.5 to reduce overfitting. The final output layer is a Dense layer with a softmax activation function, producing class probabilities corresponding to the number of land cover classes (5 classes).

In terms of implementation, transfer learning was applied by freezing all layers in the base MobileNetV2 model (layer.trainable = False), allowing the network to retain useful pre-trained features from ImageNet while training only the added top layers.

The model was compiled using the Adam optimizer with a learning rate of 1e-4, and categorical cross-entropy was used as the loss function, appropriate for multiclass classification. Training was performed for up to 100 epochs, with early stopping monitoring the validation loss and restoring the best-performing weights if no improvement was observed for 10 consecutive epochs.

This implementation leverages the compact and efficient architecture of MobileNetV2 while adapting it through transfer learning and custom top layers to suit the specific satellite image classification task.

### *2.4.3. DenseNet121 Architecture*

DenseNet121 uses a Dense Connections approach where each layer is directly connected to all its preceding layers, allowing for more effective information flow throughout the network. This approach reduces the number of parameters needed to achieve high accuracy [16]. The DenseNet121 architecture can be seen in Figure 7.

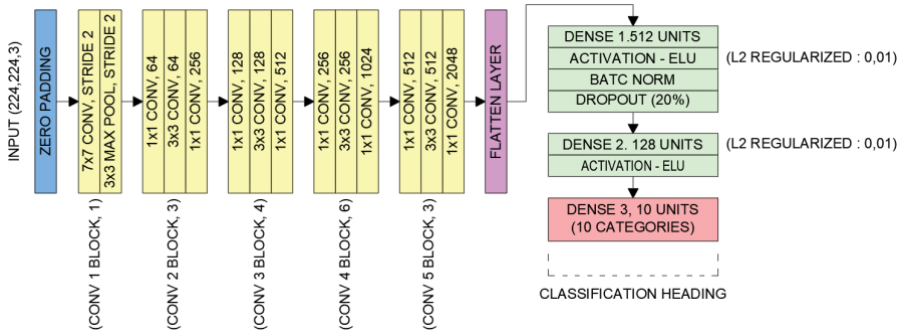


Figure 7. DenseNet121 Architecture

DenseNet121 is a convolutional neural network architecture designed to improve feature propagation and reduce the number of parameters by employing dense connectivity patterns. In this study, the architecture was implemented using an input image size of 128 × 128 × 3, adapted from the pre-trained DenseNet121 model trained on the ImageNet dataset and fine-tuned for satellite image classification.

The architecture begins with an initial convolutional layer (3×3 kernel with stride 2) preceded by zero-padding, followed by a 3×3 max pooling layer to reduce the spatial resolution. The network is composed of four main dense blocks, each consisting of multiple convolutional layers (1×1 followed by 3×3) that extract features. A key feature of DenseNet121 is its dense connections, where the output of each layer is concatenated with the outputs of all preceding layers within the same block. This dense connectivity enriches the feature representation and facilitates stronger gradient flow.

Transition layers between the dense blocks include 1×1 convolution layers and 2×2 average pooling, which serve to compress feature maps and control the network's complexity. At the end of the final dense block, a global feature map of size 4×4×1280 is produced, which is flattened in the customized model.

In the modified architecture, a fully connected layer with 256 units and ReLU activation is added to extract high-level representations, followed by a final Dense layer with softmax activation for classification into five land cover classes. To prevent overfitting and leverage prior knowledge, all layers of the base DenseNet121 model are frozen (trainable = False), allowing only the added top layers to be trained.

The model is compiled using the Adam optimizer with a learning rate of 1e-4, and the categorical crossentropy loss function, which is suitable for multi-class classification tasks. Training is carried out for up to 100 epochs, with early stopping applied (patience = 10) to restore the best-performing weights based on validation loss.

This implementation of DenseNet121 offers strong feature reuse, efficient parameter usage, and robust learning capabilities, making it well-suited for satellite image classification tasks with relatively small yet complex datasets.

### *2.5. Evaluation Model*

The confusion matrix is used in the model evaluation phase to assess the model's performance. It is a tool for evaluating the performance of a classification model by comparing the model's predictions to the actual values [34]. The matrix consists of four main components:

1. True Positive (TP): Cases where the model predicts positive and it is actually positive.
2. True Negative (TN): Cases where the model predicts negative and it is actually negative.
3. False Positive (FP): Cases where the model predicts positive but it is actually negative.
4. False Negative (FN): Cases where the model predicts negative but it is actually positive.

For multiclass classification, this concept is extended to each class, where the diagonal values in the matrix represent correct predictions (TP), and other values indicate prediction errors [34]. Several important evaluation metrics can be derived from the confusion matrix, such as:

1. Accuracy measures the percentage of correct predictions. The calculation for Accuracy is shown in Equation.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

2. Precision measures the proportion of true positive predictions among all positive predictions. The calculation for Precision is shown in Equation 4.

$$Precission = \frac{TP}{TP+FP} \quad (4)$$

3. Recall measures the model's ability to detect positive classes. The calculation for Recall is shown in Equation 5.

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

4. F1-Score is the harmonic mean of precision and recall. The calculation for F1-Score is shown in Equation 6.

$$F1 - Score = 2 \times \frac{Precission \times Recall}{Precission+Recall} \quad (6)$$

These evaluation metrics were chosen because the dataset is imbalanced, with majority classes like Forest and Settlements having around 380 samples, while minority classes such as Summit and River have fewer than 35 samples. Accuracy alone can be misleading as it may favor majority classes. Precision and recall provide insight into the model's ability to correctly identify each land cover type, especially the underrepresented ones, ensuring reliable predictions and comprehensive detection. The F1-score balances precision and recall, offering a single informative metric for overall model performance in this multi-class, imbalanced classification task.

p-ISSN: 2528-1682
e-ISSN: 2527-9165

## 3. RESULT AND DISCUSSION

This study focuses on land cover classification of the Mount Slamet region using Sentinel-2 satellite imagery and convolutional neural network (CNN) architectures, namely VGG-16, MobileNetV2, and DenseNet121. The primary problem addressed is the challenge of accurate classification across multiple land cover classes with imbalanced data distribution, particularly for underrepresented classes such as Summit and River. To overcome this, preprocessing, including cloud removal (dehazing) and data augmentation techniques, was applied, followed by a comprehensive evaluation of model performance.

### *3.1. Preprocessing*

The cloud removal (dehazing) process was performed using the Multi-Scale Fusion (MSF) method to eliminate haze and enhance the clarity of satellite images. Multi-Scale Fusion operates by estimating light transmission and applying pixel-level corrections to preserve the natural colors of the images. The results of the MSF process can be seen in Figure 8.

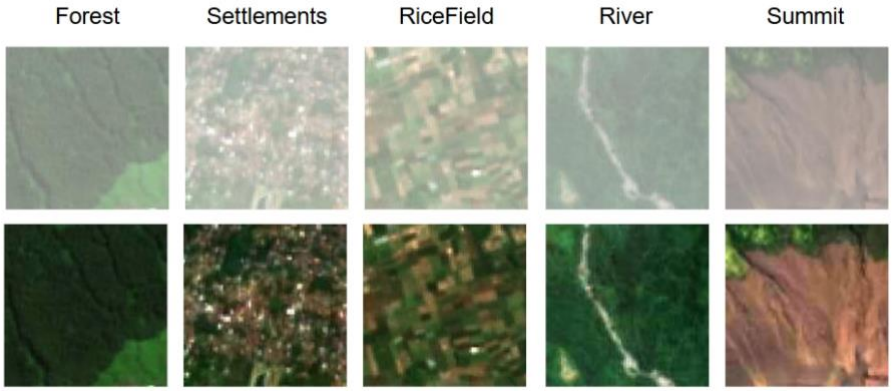


Figure 8. Dehazing Process, (Top) Before (Bottom) After

The Guided Filter is used to refine dehazed images by preserving object edges and reducing noise without eliminating important details. By combining MSF and the Guided Filter, satellite images achieve a higher level of clarity, enhancing accuracy in the land cover classification process. The results of the MSF and Guided Filter processes can be seen in Figure 9.

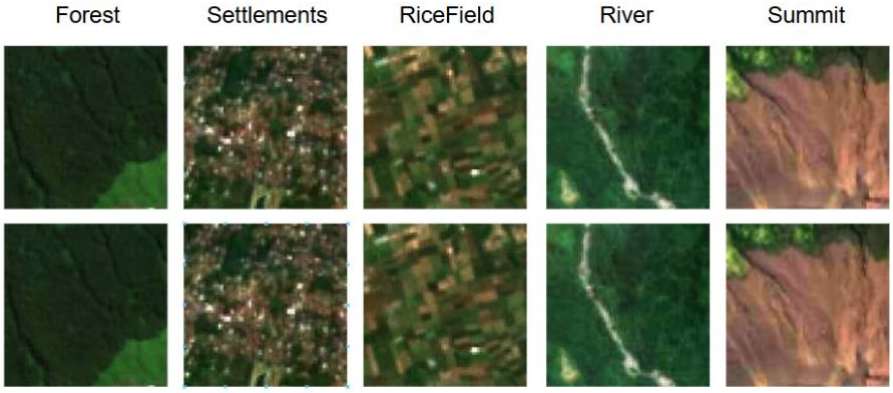


Figure 9. Guide Filter Process, (Top) Before (Bottom) After

Data imbalance among classes requires augmentation. Data augmentation is the process of creating additional data from the original dataset through various transformations to enhance diversity and size of the dataset [28]. Augmentation is applied to the Summit and River classes, which have the

fewest data points. The augmentation process includes mirror_horizontal, mirror_vertical, and rotated_180°. The number of datasets after augmentation is 1275 data points, with details presented in Table 2. This augmentation is essential for enabling the models to learn representative features from minority classes and reduce bias toward majority classes.

Table 2. Distribution of Data After Augmentation

| Class | Number of Samples |
|---|---|
| Forest | 380 |
| Settlements | 390 |
| Summit | 124 |
| RiceField | 273 |
| River | 108 |

### *3.2. Evaluation Model*

The evaluation process involves splitting the dataset into two parts: training data and validation data. This split is automatically performed by the ImageDataGenerator function using the parameter validation_split=0.2, where 80% of the data is used for training and 20% for validation. The distribution of training and validation data is shown in Table 3.

Table 3. Distribution of Dataset

| Data | Number |
|---|---|
| Training | 1022 |
| Validation | 253 |

The training parameters used include the Adam optimizer with a learning rate of 0.0001, the categorical_crossentropy loss function for multi-class classification, and the training process was conducted over 100 epochs using EarlyStopping to halt training early.

The training results showed that VGG-16 stopped at epoch 27, MobileNetV2 at epoch 28, and DenseNet121 at epoch 32. The VGG-16 model achieved an overall accuracy of 83.00%, with its best performance in the Summit class (f1-score 0.9057), despite having the fewest samples. However, the RiceField and River classes faced challenges with recall values of 0.5185 and 0.4762, respectively. Meanwhile, MobileNetV2 recorded the highest accuracy at 85.38%, excelling in the Forest (f1-score 0.9259) and Summit (f1-score 0.9388) classes, though it still struggled with significant misclassifications in the RiceField and River classes. The DenseNet121 model achieved an accuracy of 83.79%, with a perfect f1-score of 1.000 in the Summit class, demonstrating its advantage in handling classes with a small number of samples. However, DenseNet121's performance declined in the Settlements (f1-score 0.7586) and RiceField (f1-score 0.7333) classes, reflecting challenges in maintaining a balance between precision and recall in certain classes. The test results for each architecture are presented in Table 4.

Tabel 4. Evaluation Results

| Model | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| VGG-16 | 0.8300 | 0.8419 | 0.8300 | 0.8147 |
| MobileNetV2 | 0.8538 | 0.8546 | 0.8538 | 0.8486 |
| DenseNet121 | 0.8379 | 0.8438 | 0.8379 | 0.8371 |

The performance comparison of various models based on four key evaluation metrics: Accuracy, Precision, Recall, and F1-Score is presented in Figure 10.

p-ISSN: 2528-1682
e-ISSN: 2527-9165

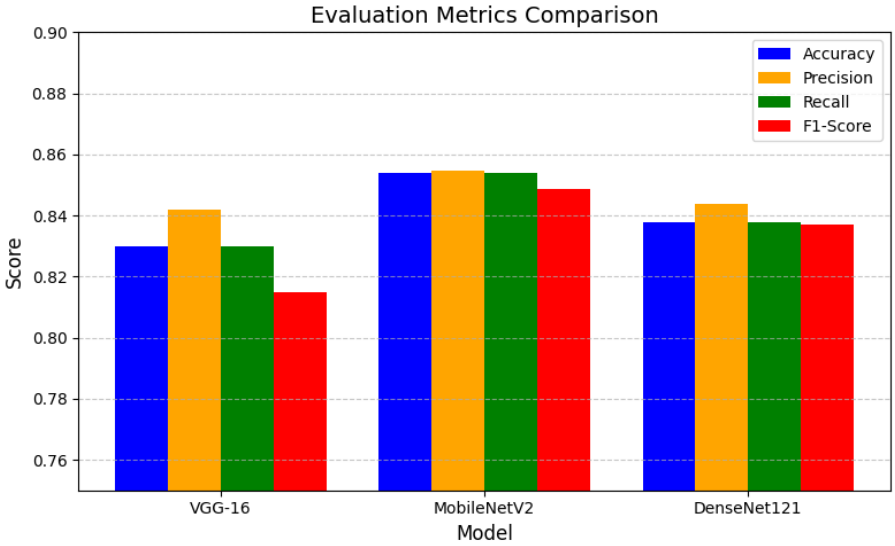


Figure 10. Comparison of Each Model's Performance

### *3.2. Discussion*

Similar studies have been conducted by Geetha et al., [18], who implemented CNN architectures such as VGG-16, ResNet34, and a custom model on Sentinel-2 satellite imagery in the Davanagere District. Their dataset consisted of 18,000 images evenly distributed across six classes, yielding high accuracies—90.7% with VGG-16, 91.5% with ResNet34, and 93.73% with a custom model. In contrast, this study focuses on land cover classification in the Mount Slamet region, which presents distinct challenges, including limited and imbalanced data, as well as varying haze levels due to mountainous conditions. Despite these constraints, the CNN models used in this study—VGG-16, MobileNetV2, and DenseNet121—achieved respectable accuracies of 83.0%, 85.4%, and 83.8%, respectively.

These results demonstrate that with appropriate preprocessing (MSF and Guided Filter) and targeted augmentation, CNNs can still perform well in complex and limited data scenarios. The strong performance in the Summit class across all models further suggests the potential of CNNs to generalize even in low-sample situations when preprocessing is handled properly.

However, several limitations should be noted. First, the dataset was relatively small compared to similar studies, which may limit the generalizability of the findings. Second, while augmentation helps mitigate class imbalance, it may not fully represent the natural variability within underrepresented classes. Additionally, only three CNN architectures were explored, all of which were pre-trained models without fine-tuning for domain-specific features.

Future research could expand the dataset both in size and geographic diversity, explore more advanced or hybrid architectures, and integrate temporal features to analyze seasonal changes. Incorporating semantic segmentation or object detection approaches could also provide more detailed spatial classification results beyond image-level predictions.

## 4. CONCLUSION

This study proposed a land cover classification approach for Sentinel-2 satellite imagery over the Slamet Mountain region using Convolutional Neural Network (CNN) architectures, addressing key challenges such as haze interference and class imbalance. To mitigate these issues, preprocessing techniques were applied, including Multi-Scale Fusion (MSF) for haze removal and a Guided Filter for image smoothing. Three CNN architectures—VGG-16, MobileNetV2, and DenseNet121—were trained and evaluated.

The experimental results demonstrated that MobileNetV2 achieved the highest accuracy of 85.4%, followed by DenseNet121 (83.8%) and VGG-16 (83.0%). These results highlight MobileNetV2's

capability to effectively handle imbalanced data and visual degradation caused by haze. Furthermore, the applied preprocessing methods significantly enhanced image quality, particularly for classes with limited sample sizes, and contributed to overall classification accuracy.

The main contribution of this study lies in integrating haze reduction and CNN-based classification for satellite imagery in mountainous regions, which are typically more challenging due to atmospheric and topographic variability. This demonstrates the effectiveness of combining image enhancement techniques with lightweight deep learning models for remote sensing applications in complex environments.

For future research, it is recommended to expand the dataset both spatially and temporally, adopt generative augmentation techniques such as GANs to address class imbalance, and explore more advanced CNN architectures or hybrid architectures. Additionally, incorporating semantic segmentation approaches could further improve spatial precision in land cover classification tasks.

## REFERENCES


[1] R. W. Van Bemmelen, "The Geology of Indonesia. General Geology of Indonesia and Adjacent Archipelagoes," 1949.

[2] I. Maryanto, M. Noerdjito, and T. Partomihardjo, *Ekologi Gunung Slamet*, no. January 2012. 2016.

[3] A. C. Anissa, E. F. Rini, and S. Soedwiwahjono, "Analisis perbandingan perubahan tutupan lahan menggunakan Citra Satelit Landsat 8 di Kecamatan Tawangmangu," *Reg. J. Pembang. Wil. dan Perenc. Partisipatif*, vol. 19, no. 1, p. 184, 2024, doi: 10.20961/region.v19i1.66929.

[4] Basuki, B. Hermiyanto, and S. A. Budiman, "Identifikasi Dan Estimasi Kerusakan Tanah Dengan Metode Berbasis Obia Citra Satelit Sentinel-2b Dan Pembobotan Lereng Gunung Raung," vol. 11, no. 1, pp. 56–72, 2023, doi: 10.29303/jrpb.v11i1.443.

[5] B. Aprilia, "Analisis Perubahan Penggunaan Lahan Kawasan Lindung Sekitar Gunung Slamet, Kabupaten Banyumas Tahun 2008 & 2019," pp. 2019–2020, 2020.

[6] J. R. Jensen, *Introductory Digital Image Processing A Remote Sensing Perspective*. 2015.

[7] M. Fayaz, J. Nam, L. M. Dang, H. Song, and H. Moon, "Land-Cover Classification Using Deep Learning with High-Resolution Remote-Sensing Imagery," no. Lc, pp. 1–15, 2024.

[8] S. Talukdar *et al.*, "Land-Use Land-Cover Classification by Machine Learning Classifiers for Satellite Observations—A Review," 2020.

[9] I. Daqiqil, "Machine Learning: Teori, Studi Kasus, dan Implementasi Menggunakan Phyton," 2021.

[10] C. A. S. Kinasih and H. Hidayat, "Ekstraksi Data Bangunan Dari Data Citra Unmanned Aerial Vehicle Menggunakan Metode Convolutional Neural Networks (CNN) (Studi Kasus: Desa Campurejo, Kabupaten Gresik)," *Geoid*, vol. 17, no. 1, p. 81, 2022, doi: 10.12962/j24423998.v17i1.10289.

[11] R. Venkatesan and B. Li, *Convolutional Neural Networks in Visual Computing : A Concise Guide*. 2017.

[12] Q. Liu, B. Wang, S. Tan, S. Zou, and W. Ge, "Remote Sensing Image Dehazing Using Multi-Scale Gated Attention for Flight Simulator," *IEICE Trans. Inf. Syst.*, vol. E107.D, no. 9, pp. 1206–1218, 2024, doi: 10.1587/transinf.2023EDP7191.

[13] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: BEYOND EMPIRICAL RISK MINIMIZATION," *Iclr*, no. March, pp. 1–8, 2018.

[14] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition," in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 512–519. doi: 10.1109/CVPRW.2014.131.

[15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520. doi: 10.1109/CVPR.2018.00474.

[16] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2261–2269. doi: 10.1109/CVPR.2017.243.

[17] P. Helber, B. Bischke, A. Dengel, and D. Borth, "EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 12, no. 7, pp. 2217–2226, 2019, doi: 10.1109/JSTARS.2019.2918242.

[18] Geetha, A. G. Karegowda, Nandeesha, and N. B. V, "Classification of Sentinel 2 Images using Customized Convolution Neural Networks," *Orig. Res. Pap. Int. J. Intell. Syst. Appl. Eng. IJISAE*, vol. 2023, no. 1s, pp. 136–142, 2023, [Online]. Available: www.ijisae.org

[19] M. Ba, P. I. Thiam, E. Delay, C. A. Ngom, I. Diop, and A. Bah, "Deep Learning-based Land Use and Land Cover Changes Detection from Satellite Imagery : a case study of the city of Richard Toll," *ACM Int. Conf. Proceeding Ser.*, pp. 60–68, 2024, doi: 10.1145/3653946.3653956.

[20] S. Basheer, X. Wang, R. A. Nawaz, T. Pang, T. Adekanmbi, and M. Q. Mahmood, "A comparative analysis of PlanetScope 4-band and 8-band imageries for land use land cover classification," *Geomatica*, vol. 76, no. 2, p. 100023, 2024, doi: 10.1016/j.geomat.2024.100023.

[21] S. S. Burrewar, M. Haque, and T. U. Haider, "Convolutional Neural Network Methods for Detecting Land-Use Changes," *Int. J. Intell. Syst. Appl. Eng.*, vol. 12, no. 14s, pp. 573–590, 2024.

[22] ESA, "Sentinel-2 Mission (European Space Agency)." [Online]. Available: https://sentinel.esa.int/web/sentinel/missions/sentinel-2

[23] A. Asokan, J. Anitha, M. Ciobanu, A. Gabor, A. Naaji, and D. J. Hemanth, "Image processing techniques for analysis of satellite images for historical maps classification-An overview," *Appl. Sci.*, vol. 10, no. 12, 2020, doi: 10.3390/app10124207.

[24] L. A. Santos, K. R. Ferreira, G. Camara, M. C. A. Picoli, and R. E. Simoes, "Quality control and class noise reduction of satellite image time series," *ISPRS J. Photogramm. Remote Sens.*, vol. 177, no. May, pp. 75–88, 2021, doi: 10.1016/j.isprsjprs.2021.04.014.
[25] M. Yao, Q. Miao, and Q. Hao, "Image Dehazing Method Based on Multi-scale Feature Fusion," vol. 119, no. Essaeme, pp. 2163–2166, 2017, doi: 10.2991/essaeme-17.2017.438.
[26] A. Codruta and C. Ancuti, "Single Image Dehazing by Multi-Scale Fusion," *IEEE Trans. Image Process.*, vol. 22, 2013, doi: 10.1109/TIP.2013.2262284.
[27] K. He, J. Sun, and X. Tang, "Guided Image Filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, 2013, doi: 10.1109/TPAMI.2012.213.
[28] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J. Big Data*, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0197-0.
[29] L. Perez and J. Wang, "The Effectiveness of Data Augmentation in Image Classification using Deep Learning," 2017.
[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. Burges, L. Bottou, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf
[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.
[32] C. Szegedy *et al.*, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9. doi: 10.1109/CVPR.2015.7298594.
[33] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," 2017, [Online]. Available: http://arxiv.org/abs/1704.04861
[34] D. M. W. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," pp. 37–63, 2011, [Online]. Available: http://arxiv.org/abs/2010.16061