

Article

DeepDR: A Two-Level Deep Defect Recognition Framework for Meteorological Satellite Images

Xiangang Zhao ¹, Xiangyu Chang ¹, Cunqun Fan ¹, Manyun Lin ^{1,*}, Lan Wei ¹ and Yunming Ye ²

¹ National Satellite Meteorological Center, Beijing 100086, China; xgzhaocma.gov.cn (X.Z.); changxy@cma.gov.cn (X.C.); fancq@cma.gov.cn (C.F.); weilan@cma.gov.cn (L.W.)

² The Department of Computer Science, Harbin Institute of Technology, Shenzhen 518055, China; yeyunming@hit.edu.cn

* Correspondence: linmy@cma.gov.cn

Abstract: Raw meteorological satellite images often suffer from defects such as noise points and lines due to atmospheric interference and instrument errors. Current solutions typically rely on manual visual inspection to identify these defects. However, manual inspection is labor-intensive, lacks uniform standards, and is prone to both false positives and missed detections. To address these challenges, we propose DeepDR, a two-level deep defect recognition framework for meteorological satellite images. DeepDR consists of two modules: a transformer-based noise image classification module for the first level and a noise region segmentation module based on a pseudo-label training strategy for the second level. This framework enables the automatic identification of defective cloud images and the detection of noise points and lines, thereby significantly improving the accuracy of defect recognition. To evaluate the effectiveness of DeepDR, we have collected and released two satellite cloud image datasets from the FengYun-1 satellite, which include noise points and lines. Subsequently, we conducted comprehensive experiments to demonstrate the superior performance of our approach in addressing the satellite cloud image defect recognition problem.

Keywords: satellite cloud images; defect image classification; noise region segmentation; pseudo-labels



Academic Editors: Salah Bourennane and Kevin Tansey

Received: 4 January 2025

Revised: 27 January 2025

Accepted: 27 January 2025

Published: 8 February 2025

Citation: Zhao, X.; Chang, X.; Fan, C.; Lin, M.; Wei, L.; Ye, Y. DeepDR: A Two-Level Deep Defect Recognition Framework for Meteorological Satellite Images. *Remote Sens.* **2025**, *17*, 585. <https://doi.org/10.3390/rs17040585>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, advancements in launch technology have facilitated the deployment of multiple meteorological satellites, such as the Fengyun meteorological satellites. These satellites are equipped with various sensors to gather information about the Earth's atmosphere, providing comprehensive, timely, and dynamic cloud imagery. The imagery captured by these satellites serves a multitude of purposes. For instance, Kim et al. [1] explored the utilization of meteorological satellite images for short-term photovoltaic power forecasts. These images offer a top-down perspective of the atmosphere and local environment, aiding in the monitoring of climate change and solar radiation. Similarly, Vyas et al. [2] utilized data from Indian geostationary satellites, Kalpana-1 and INSAT 3A, to develop an Early Warning Indicator for agricultural drought. Hence, meteorological satellite images play a vital role in various tasks such as weather forecasting, disaster prediction, and supporting agricultural production.

However, the space environment is susceptible to electromagnetic interference, making satellite sensors vulnerable to disruptions. This vulnerability results in collected images containing various types of noise that significantly reduce their usability. These defects

are mainly attributed to atmospheric interference and instrument errors, among other factors. In this paper, we categorize defective satellite cloud images into two types: noise point images and noise line images. Figure 1 illustrates examples of satellite cloud images with noise points and lines. Figure 1a,b display satellite cloud images with noise points resembling salt and pepper noise, caused by factors such as image sensors, transmission channels, and decoding processes. Figure 1c,d show images with noise lines, which typically appear as black or white rectangular strips due to interference or faults in the imaging process.

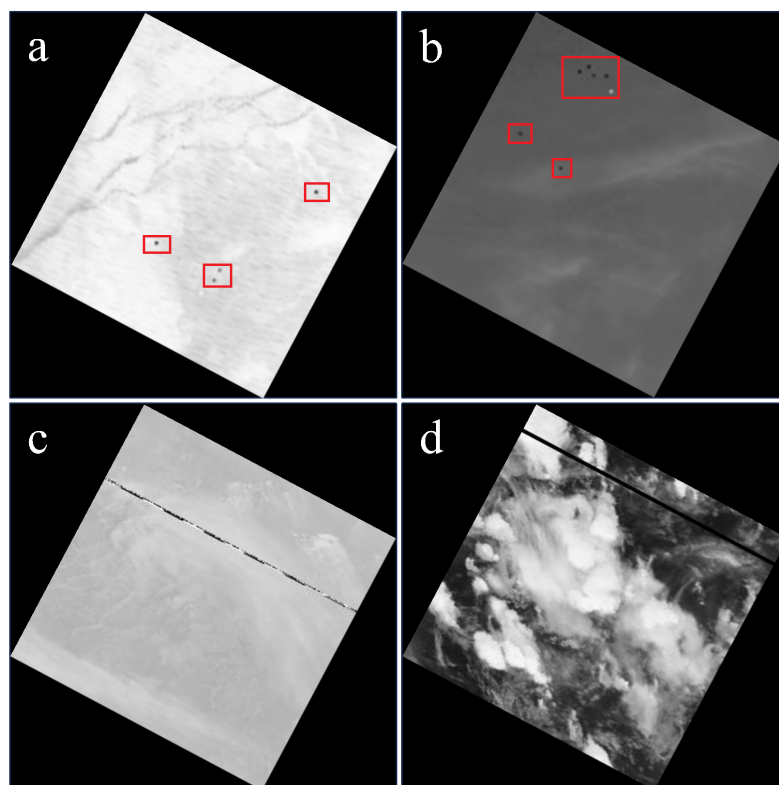


Figure 1. Some examples of satellite cloud images with noise points and lines are presented. The noise points have been marked with red boxes. In (a,b), satellite cloud images with noise points are displayed, while (c,d) show images with noise lines.

Defective satellite cloud images can affect the performance of models in downstream tasks. For example, in agricultural applications, satellite sensors can collect various spectral characteristics of different crops at different stages based on biological principles. This information is used to identify the growth status, crop categories, surface information, and estimate parameters such as crop area and yield per unit area. However, the presence of defects directly affects tasks such as estimating crop area and identifying surface information, reducing the accuracy of satellite image applications in agriculture. Therefore, accurate identification of defective images, including detecting the regions of interest, is crucial for many applications based on meteorological satellite images.

Existing defect recognition methods for satellite cloud images mainly rely on manual visual inspection and have not yet achieved automated recognition. Manual inspection methods are often cumbersome, lack standardized procedures, exhibit low accuracy, and are prone to issues such as false positives and false negatives. Therefore, in this paper, we propose a novel two-level deep defect recognition framework, namely DeepDR, for meteorological satellite images. This framework can automatically identify defective cloud images and detect noise points and lines, significantly enhancing the accuracy and efficiency of defect recognition. The two-level framework comprises a transformer-based noise image

classification module and a noise region segmentation module, corresponding to the first and second levels, respectively. In the first level, the noise image classification module is utilized to determine whether a meteorological satellite image contains noise points or lines. Based on the results of image classification, the second level employs the noise region segmentation module to detect the regions of noise points and lines.

In the noise image classification module, we develop a novel transformer network aimed at learning the characteristics of defective satellite images. The transformer network is designed to capture long-range dependencies and global contextual information of satellite images through self-attention mechanisms, thus effectively identifying the classes of defective images. To evaluate the accuracy of noise image classification, we collect and release two datasets specifically for satellite cloud image classification. In the noise region segmentation module, we devise a training strategy that uses pseudo-labels to construct models. We first create two pseudo-noise datasets that mimic the characteristics of real noise points and noise lines. These datasets are utilized to train existing popular image segmentation models capable of detecting regions containing noise points or noise lines. We then assess the accuracy of the trained image segmentation models, which have been trained on pseudo-labels, using meteorological satellite images with real noise points and noise lines. Our experiments demonstrate that the proposed training strategy is effective and useful.

The contributions of this paper are summarized as follows:

- We propose a novel two-level deep defect recognition framework for meteorological satellite images. To our knowledge, this problem has not been explored previously through deep learning methods.
- We develop a transformer-based noise image classification method to identify whether a meteorological satellite image contains noise points or noise lines. Additionally, we construct and release two datasets of noise image classification to evaluate the proposed method.
- We design a training strategy using pseudo-labels to train image segmentation models to detect the region containing noise points or noise lines. This training strategy can be applied to train image segmentation models for detecting noise regions.
- Comprehensive experiments have been conducted to evaluate the proposed noise image classification method and the training strategy of noise region segmentation. The results demonstrate that our method outperforms state-of-the-art methods in addressing the noise image classification problem, and our training strategy can effectively construct image segmentation models to detect real noise regions.

The rest of this paper is organized as follows: Section 2 provides a brief review of the related work. Section 3 presents a detailed description of our method. In Section 4, we introduce the experimental results in detail. Section 5 concludes this paper with a summary.

2. Related Work

2.1. Noise Image Classification

The classification methods of noise images can be systematically categorized into three distinct methodological approaches: rule-based, machine learning-based, and deep learning-based methods.

Rule-based methods for noise image classification leverage predefined rules and thresholds to identify noise images and non-noise images. The Selective Median Filter (PSM) [3] is a prime example, which employs an empirical threshold to compare the absolute difference between the original pixel grayscale value and the median filtering value, applying weighted median filtering only when this difference exceeds the threshold. Additionally, the Tri-State Median Filter (TSM) [4] attempts to balance noise removal

with detail preservation by calculating outputs from both standard and center-weighted median filters and comparing these outputs against a fixed threshold. More sophisticated approaches, like the Adaptive Center-Weighted Median Filter (ACWM) [5] and the Relative Ordering of Absolute Differences (ROAD), iteratively refine their thresholds to adapt to varying noise conditions, enhancing their ability to detect subtle noise variations. These rule-based methods form the foundation of noise detection to classify images with and without noise. Traditional rule-based methods rely on manually designed features and thresholds, often requiring extensive domain expertise. However, their rigid structure makes them ill suited to handle complex and varying noise patterns, especially when noise characteristics differ significantly across different satellite imaging systems. These limitations have driven researchers to explore more flexible machine learning approaches, which are better equipped to handle such complex noise environments.

Transitioning from conventional rule-based techniques, machine learning-based methods [6] offer dynamic solutions by deriving insights from data features. Algorithms like RandomForest [7] and AdaBoost [8] have proven effective in classifying satellite imagery, adeptly handling complex noise patterns within diverse environmental contexts. RandomForest operates by constructing multiple decision trees during the training phase and outputting the class that is the mode of the classes of the individual trees. By aggregating predictions from numerous decision trees, it reduces the risk of overfitting and enhances the model's generalizability. As a powerful boosting type of ensemble learner, AdaBoost improves classification accuracy by combining multiple weak classifiers into a strong one. Each successive classifier in the AdaBoost sequence focuses on the instances misclassified by previous classifiers, thereby iteratively improving the model's performance. Research employing these methodologies has shown significant improvements in classification accuracy, illustrating their ability to adapt to varied noise characteristics without necessitating manual threshold adjustments or extensive rule formulations. Machine learning techniques have demonstrated significant improvements by learning patterns from data, as opposed to relying on fixed rules. Typically, these methods depend on handcrafted features, and their performance is heavily influenced by the quality of feature engineering. However, when dealing with high-dimensional satellite imagery, manually designing features often fails to capture subtle noise patterns and complex spatial relationships effectively. To address these limitations, more advanced deep learning-based approaches have been proposed.

The advent of deep learning has markedly transformed noise image classification, with models such as AlexNet [9] and ResNet [10] establishing new benchmarks in the domain. AlexNet consists of five convolutional layers followed by three fully connected layers and employs techniques such as ReLU activations and dropout to prevent overfitting. It can discern complex patterns from high-resolution images, enabling effective classification of land use and land cover from raw pixel values. It is particularly effective in scenarios where the distinction between different terrains or objects is subtle yet crucial. Benefiting from its powerful classification capability, it can be applied in many classification tasks in satellite images. ResNet introduces the concept of residual learning to address the degradation problem in very deep networks. By integrating skip connections that allow inputs to bypass one or more layers, ResNet can train much deeper networks than previously possible. This architecture has proven highly effective in satellite image classification, where deeper models are beneficial for capturing the hierarchical features inherent in complex landscapes. ResNet can effectively handle the challenge of identifying small objects or fine details in large-scale satellite images, such as roads, buildings, and various types of vegetation. It shows potential in classifying images with small noise areas. These deep learning architectures employ multi-layered neural networks to autonomously extract and assimilate features from extensive datasets, achieving unprecedented accuracy levels

in satellite image classification. The complexity and depth of these networks enable them to capture intricate patterns and distinctions in image data, frequently surpassing traditional machine learning techniques in a myriad of scenarios.

In recent years, with the advancement of remote sensing image processing technology, both noise detection and denoising techniques have become increasingly sophisticated, serving as valuable complements to noise image classification methods. Several notable approaches have emerged in noise detection: DirectNet [11] (Dual-Window-Inspired Reconstruction Network) effectively separates targets from backgrounds to identify anomalies; BockNet [12], a self-supervised blind-block network, enhances anomalous pixel detection by amplifying abnormal features; SparseHAD [13] learns discriminative latent reconstruction with minimal errors for background pixels while maximizing error detection for anomalous elements; DSLF [14] (Deep Self-representation Learning Framework) leverages spatial information through a multi-scale strategy for noise detection. While these methods face their own challenges, such as requiring large amounts of labeled training data and significant computational resources, they represent the current state of the art in addressing the noise image classification problem.

In terms of denoising approaches, Eigen-CNN [15], an eigenimage plus eigennoise level map-guided convolutional neural network, demonstrates excellent performance in HSI (Hyperspectral Image) denoising. HyADD [16], a hierarchical integration framework for hyperspectral simultaneous Anomaly Detection (AD) and denoising, innovatively integrates joint AD and denoising processes. Its iterative approach enables the mutual enhancement of outputs, overcoming the limitations of traditional two-stage schemes. Li Xian et al. [17] proposed a unified deep learning framework for joint denoising and classification of high-dimensional images, specifically applied to hyperspectral imaging frameworks. This approach yields impressive denoising results as a beneficial by-product of the classification process.

2.2. Image Segmentation

In the field of image segmentation, recent advancements have been marked by significant developments across various architectures, each tailored to specific types of imagery and segmentation tasks. One of the foundational architectures in medical image segmentation is the U-Net, introduced by Ronneberger et al. [18], which features a contracting path to capture context and a symmetric expanding path for precise localization. This model has been pivotal in setting standards due to its efficiency and accuracy in medical applications. Building upon the success of U-Net, Zhou et al. [19] developed UNet++, which enhances the original structure with nested, dense skip pathways. These modifications aim to improve gradient flows and feature propagation, enabling the architecture to handle fine details more effectively.

Parallel to the developments in medical imaging, Chen et al. [20,21] introduced DeepLabV3 [20] and its extension DeepLabV3+ [21]. These models utilize atrous convolution to robustly manage scale variations and incorporate an atrous spatial pyramid pooling (ASPP) and an encoder–decoder structure. These features make DeepLabV3 and DeepLabV3+ highly suitable for complex semantic segmentation tasks, providing enhanced capability for boundary delineation and multi-scale information processing.

Additionally, the field of remotely sensed image segmentation has also seen innovative contributions. Here, the MACU-Net [22] addresses the challenges of fine-resolution remotely sensed images by employing a multi-scale architecture that effectively handles various image resolutions, proving beneficial in environmental and geographical applications. Similarly, the A2-FPN model [23] enhances feature pyramid networks to optimize the segmentation of high-resolution satellite imagery, focusing on detailed geographical features.

SWINT-RESNet [24], based on multi-feature fusion, combines transformer-extracted global and local features with CNN features to enhance the accuracy of remote sensing image segmentation. AFENet [25] achieves robust feature extraction through optimized channel reduction, scale expansion, and channel redistribution operations. X Kang et al. [26] proposed a hierarchical class tree for high-resolution remote sensing image semantic segmentation, which demonstrates superior segmentation performance.

These contributions collectively underscore the dynamic evolution of image segmentation technologies, showing a trend towards more specialized and refined approaches that cater to the distinct demands of the image segmentation task.

3. Deep Defect Recognition

3.1. Problem Definition

Generally, the size of each satellite image is relatively large, and the noise points and lines in satellite images are distributed widely and appear very sparse. If we directly train a model to identify noise points and lines in the image, it is impractical. Firstly, building such a model is challenging. The model needs to identify sparse noise points and lines in an image with such high resolution, demanding strong feature extraction capabilities. Secondly, it wastes a considerable amount of computational resources. The model needs to calculate a large number of normal image portions during the feature extraction process, which is unnecessary. Thirdly, simply classifying the image is not practically meaningful. Although it is easy to achieve noise point and line classification for the original image, it has broad coverage and cannot provide accurate localization.

Hence, we change the original defect recognition problem of noise point and noise line into a binary classification problem and a further image segmentation problem of smaller satellite images. Specifically, we define the original satellite image as $S \in \mathcal{R}^{m \times n \times b}$, where $m \times n$ is its size and b is the number of channels. Firstly, we crop each satellite image into many smaller images $\{X_1, X_2, \dots\}$ and the size of $X_i \in \mathcal{R}^{r \times r \times b}$ is $r \times r$. In addition, we train a transformer-based image classification model $f_\theta(\cdot)$ to determine whether X_i is an image with noise points/lines. Finally, we design a pseudo-label-based region segmentation model to select the noise points/lines in these images.

3.2. Overall Framework

In this work, we propose a novel two-level deep defect recognition framework, called DeepNR, to identify noise point or noise line images and detect corresponding noise regions for meteorological satellite images. The overall framework is shown in Figure 2 and the pseudo-code is shown in Algorithm 1. First and foremost, the original satellite image S can be cropped into many smaller images X_i , which are denoted as the samples of the model. In addition, for each sample X , DeepNR designs a novel classifier called Transformer-based Noise Image Classifier to determine whether it is a normal image. In other words, this framework designs a noise point classifier and a noise line classifier to identify images with noise points and noise lines, respectively. Last but not least, DeepNR is designed with a pseudo-labels training strategy to train region segmentation models to detect the noise points/lines regions in these samples. These region segmentation models can accurately identify which pixel is defective.

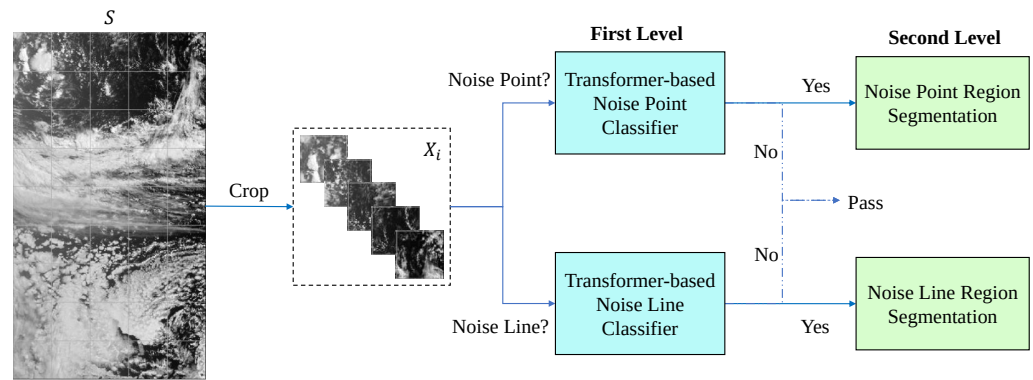


Figure 2. Illustration of the proposed framework DeepDR.

Algorithm 1 DeepDR: Two-level Deep Defect Recognition Framework

Data: Original satellite image S

Results: Defect detection results (normalPatches, pointMasks, lineMasks)

```

1: patches  $\leftarrow$  CropImage( $S$ , patchSize);
2: noisePointPatches  $\leftarrow$  { };
3: noiseLinePatches  $\leftarrow$  { };
4: normalPatches  $\leftarrow$  { };
5: for each patch  $X_i$  in patches do
6:   isNoisePoint  $\leftarrow$  TransformerClassifierPoint( $X_i$ );
7:   if isNoisePoint then
8:     noisePointPatches.append( $X_i$ );
9:     continue;
10:  else
11:    isNoiseLine  $\leftarrow$  TransformerClassifierLine( $X_i$ );
12:    if isNoiseLine then
13:      noiseLinePatches.append( $X_i$ );
14:      continue;
15:    else
16:      normalPatches.append( $X_i$ );
17:    end if
18:  end if
19: end for
20: pointMasks  $\leftarrow$  { };
21: lineMasks  $\leftarrow$  { };
22: for patch in noisePointPatches do
23:   pointMask  $\leftarrow$  NoisePointSegmentation(patch);
24:   pointMasks.append(pointMask);
25: end for
26: for patch in noiseLinePatches do
27:   lineMask  $\leftarrow$  NoiseLineSegmentation(patch);
28:   lineMasks.append(lineMask);
29: end for
30: return normalPatches, pointMasks, lineMasks
  
```

3.3. Transformer-Based Noise Image Classification

Different from existing studies, Transformer-based noise image classification (TNIC) is based on the Transformer architecture instead of ResNet. Compared to ResNet, the architecture of the Transformer leverages self-attention mechanisms, making it easier to capture global dependencies while establishing local dependencies. The overall procedure of TNIC is shown in Figure 3. The Transformer architecture used in TNIC is DeiT [27], which exhibits higher efficiency and greater robustness compared to the original architectures. The DeiT (Data-efficient Image Transformer) architecture centers on the transformer

encoder and introduces innovative elements such as the classification token and distillation token. The network begins by dividing the input image into fixed-size patches (e.g., 16×16). These patches are linearly projected into a high-dimensional space to form patch embeddings, which are then fed into a stacked transformer encoder for feature extraction. The output consists of a series of tokens, including a classification token and a distillation token. The distillation token is specifically designed to receive supervisory signals from a teacher network. During training, there are two branches: one for the pretrained teacher network and the other for the trainable backbone network. These branches interact via their respective token outputs, facilitating effective knowledge distillation.

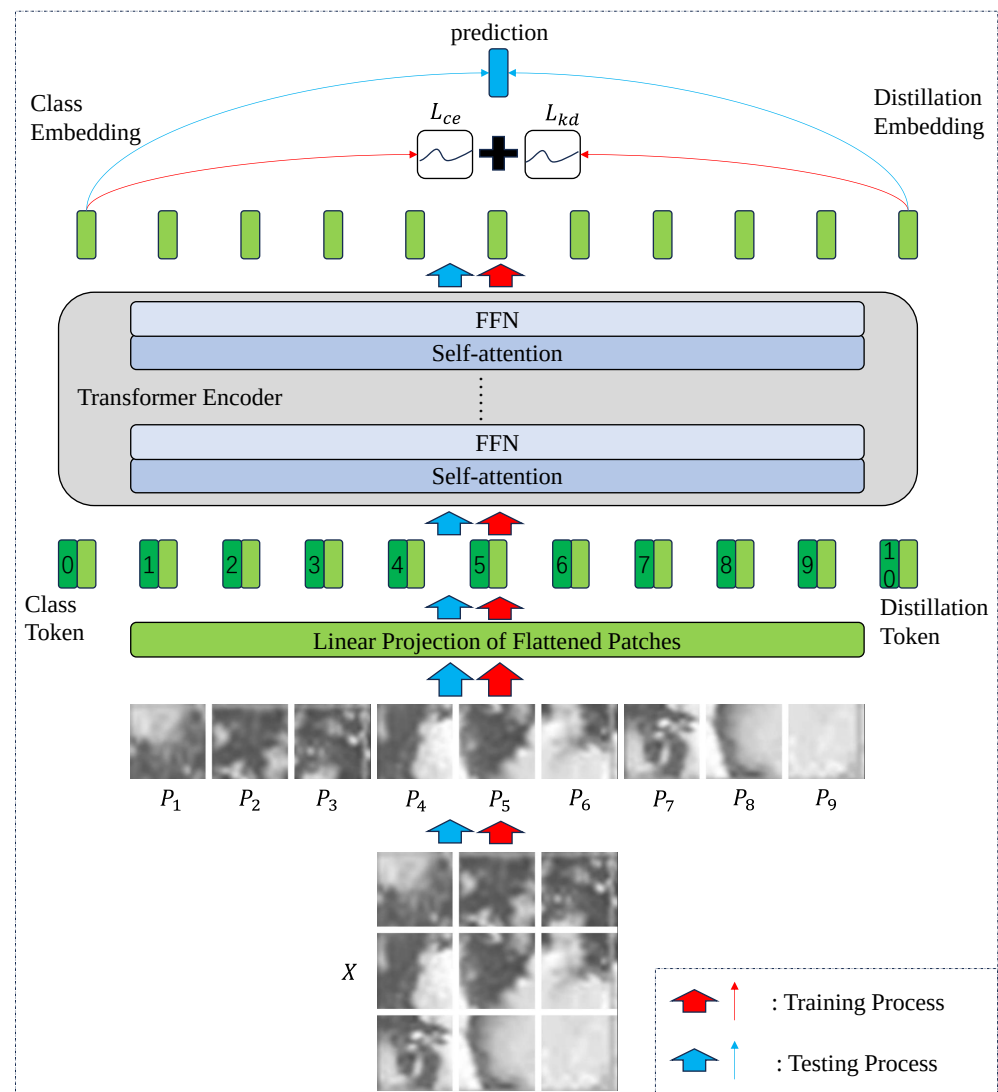


Figure 3. Illustration of the Transformer-based noise image classifier.

Satellite images contain rich spatial and temporal information, typically characterized by high resolution and high dimensionality. These images often exhibit complex spatial and temporal dependencies, which are challenging for traditional convolutional neural networks (CNNs) to model effectively. As a Transformer-based architecture, DeiT excels at capturing long-range dependencies and distant spatial relationships within the image. It employs a self-attention mechanism that allows global information exchange between different regions of the input image, helping to capture relationships between various parts of the image. This is especially crucial for defect detection in satellite images, where noise is often scattered irregularly and requires global context for accurate identification. One key

advantage of DeiT is its relatively lightweight design compared to traditional Transformer models, making it computationally efficient while still maintaining strong performance in capturing global patterns and structures. This combination of global information capture and efficiency significantly enhances detection accuracy.

The procedure of TNIC consists of a training process module (red) and a testing process module (blue). The former is used to adapt the model to the data patterns of satellite images, while the latter is employed to detect noise points and lines in satellite images.

Self-attention is a key operation in the transformer encoder. It is a special case of attention where the keys K are equal to the values V . The important layer in each module of the transformer encoder is an attention mechanism like [28]. To form the layer output, results from each self-attention are concatenated and transformed by a parameterized linear. Given a hidden matrix $X \in \mathbb{R}^{T \times M}$ as an input sequence, the mechanism first transforms it into queries $Q \in \mathbb{R}^{T \times M}$, keys $K \in \mathbb{R}^{T \times M}$, and values $V \in \mathbb{R}^{T \times M}$:

$$\begin{bmatrix} Q, K, V \end{bmatrix} = X \begin{bmatrix} W_Q, W_K, W_V \end{bmatrix}. \quad (1)$$

where $\{W_Q, W_K, W_V\} \in \mathbb{R}^{M \times M}$ are trainable parameters, and M indicates the number of variables. Then, we apply the scaled dot-product attention to calculate the weight value of every position as follows:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V. \quad (2)$$

where d is the scaling factor, used to push the softmax function into regions where it has extremely small gradients.

During the training process, the sample X obtains the hidden vectors corresponding to the class token and the distillation token through DeiT for calculating the loss function. Specifically, the original samples X are divided into multiple patches $\{P_1, P_2, \dots\}$, which are flattened and concatenated by linear projection of flattened patches to serve as the input for the transformer encoder. It is important to note that, in addition to these patch tokens, DeiT also introduces additional class tokens and distillation tokens to integrate information from these patch tokens. Using the transformer encoder, we can obtain the class embedding z_c and the distillation embedding z_d , which are hidden vectors of the class token and the distillation token, respectively. The class embedding z_c is used to calculate the cross-entropy loss function L_{ce} while the distillation embedding z_d is used to calculate the distillation loss function L_{kd} . Their combination is the final loss in training the model:

$$L = (1 - \lambda)L_{ce} + \lambda\gamma^2L_{kd}, \quad (3)$$

where λ and γ are scale factors to control the weight of two loss functions. Moreover, L_{kd} calculates the Kullback–Leibler divergence between z_d and the distillation embedding z_d^t of the teacher model.

During the testing process, the sample X obtains the hidden vectors corresponding to the class token and the distillation token through DeiT for recognizing the samples with noise points/lines. Using the transformer encoder, we can also obtain the class embedding z_c (c is for category) and the distillation embedding z_d . The model takes their combination as the final prediction to classify the samples as noise point images or noise line images.

$$y^* = \arg \max_c (z_d + z_c) \quad (4)$$

3.4. Pseudo-Label-Based Noise Region Segmentation

Noise region segmentation mainly identifies the regions in the satellite images where the pixels are either noise points or noise lines. Typically, this segmentation task employs semantic segmentation networks, which take a satellite image as input and produce a segmented image as output in the form of a binary mask. However, training these semantic segmentation networks effectively requires a significant amount of annotated data, which can be expensive and labor-intensive. Unlike simpler tasks like image classification, where assigning a single label to the entire image suffices, image segmentation demands more detailed annotations that precisely outline the boundaries between different noise regions. This annotation process can be time-consuming, particularly for complex images containing small noise points and intricate noise lines. Moreover, ensuring consistency and accuracy across annotations poses a challenge, as different annotators may interpret boundaries differently or make errors during labeling. Annotating meteorological satellite images, in particular, may necessitate specialized domain knowledge to ensure accurate labeling, further complicating the annotation task. Overall, the inherent difficulty in accurately delineating noise points and lines makes data labeling for image segmentation a challenging endeavor.

To address the challenge of labeling noise regions, we propose a training strategy that leverages pseudo-labels to generate synthetic noise satellite images and their corresponding masks. By emulating the characteristics of real noise points and lines, we randomly insert noise points and lines into normal meteorological satellite images. For the noise points, we randomly generate between two and seven points per image, with each point having a radius of 1 pixel and placed within the image boundaries. The position of each noise point is selected randomly, ensuring variability in their distribution. For the line noise, we generate up to three lines per image, with each line having a width between 1 and 3 pixels and a length ranging from 20 to 200 pixels. The lines can be either horizontal or vertical, randomly oriented within the image.

This approach enables us to construct extensive datasets of pseudo-labeled noise images, which serve as valuable resources for training semantic segmentation models. The generated datasets contain a diverse set of noise patterns, closely mimicking real-world noise phenomena. We will share the specific details of the noise generation strategy in the source code, which will provide full transparency on the parameters and methodology used in the generation process.

Figure 4 illustrates the process of pseudo-label-based noise region segmentation. Training strategies based on pseudo-labels are employed to generate noise images and masks. The noise images serve as inputs to region segmentation models, which utilize segmentation networks, typically adopting the encoder–decoder architecture as depicted in Figure 4. Here, we choose a classic model U-Net++ to explain the selection of the segmentation model: The U-Net++ architecture builds upon the classic U-Net encoder–decoder structure. The encoder extracts features from the input image, while the decoder progressively restores spatial resolution to generate segmentation outputs. Unlike the traditional U-Net, U-Net++ incorporates Dense Skip Connections to enhance feature propagation and multi-scale information fusion. These skip connections are designed as a set of dense sub-networks, where each sub-network contains multiple convolutional layers, enabling improved performance in segmentation tasks through better utilization of hierarchical feature representations.

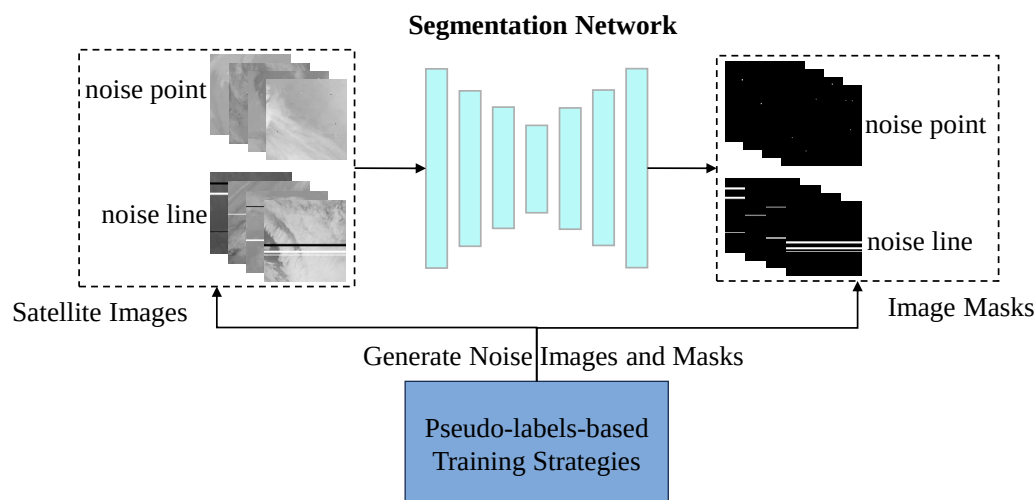


Figure 4. Illustration of the proposed pseudo-label-based noise region segmentation.

In this architecture, the encoder extracts high-level features from the input image through a series of convolutional layers. These layers progressively reduce spatial dimensions while increasing the depth of feature maps, capturing abstract representations of the input image. Following the encoder, the decoder upsamples the feature maps to the original input resolution using transposed convolutional layers, gradually recovering spatial information lost during the encoding stage. Additionally, skip connections are often incorporated between corresponding encoder and decoder layers to preserve fine-grained details and alleviate the vanishing gradient problem. These connections allow the decoder to access both low-level and high-level features, facilitating precise segmentation.

The final output of the decoder is a pixel-wise prediction map, where each pixel is assigned a semantic label indicating the class it belongs to. Through training with pseudo-labels, the network learns to accurately segment noise points and noise lines in meteorological satellite images. Leveraging the pseudo-label-based training strategy, existing image segmentation networks can be effectively combined to build noise region segmentation models.

4. Experiments

4.1. Dataset

In the experiments, our main focus was to classify defective satellite cloud images and identify the corresponding region of noise points and lines. However, there is a lack of publicly available datasets specifically designed to recognize defective satellite images. To address this gap, we carefully created a dataset using satellite cloud imagery from the Fengyun-1 satellite (China's first-generation sun-synchronous orbiting meteorological satellites), which can be downloaded from <https://satellite.nsmc.org.cn/>. The dataset consists of 20,000 images, each with a resolution of 224×224 pixels.

First, we split the dataset into training and testing subsets. The training dataset, used for model training, consists of 16,000 images, while the testing dataset, used for method evaluation, contains 4000 images. To ensure the authenticity of model evaluation, the testing dataset uses the original images, with noise manually annotated in the images, including ground truth for both detection and segmentation tasks. For the training dataset, noise is simulated by generating noise points and lines, with noise masks directly used as ground truth, enabling the rapid construction of datasets for detection and segmentation tasks. Additionally, for a more detailed evaluation, we divided the dataset into two subsets,

each focusing on a specific aspect of the defects: noise points and noise lines. Both subsets contain an equal number of samples, with 10,000 images in each.

Figure 5 illustrates examples from the noise point dataset, showing satellite cloud images with noise points on the top and normal satellite cloud images on the bottom. The noise points appear as salt and pepper noise, characterized by the sudden appearance of black or white pixels. This type of low-grayscale noise can be caused by various factors such as image sensors, transmission channels, and decoding processes.

Figure 6 presents representative samples from the noise lines dataset, with the upper segment showing satellite cloud images with noise lines and the lower segment showing normal satellite cloud images. Noise lines in satellite images typically appear as black or white rectangular strips, either horizontally or vertically on the sides of the image. These artifacts are commonly caused by interference or faults in the imaging process.

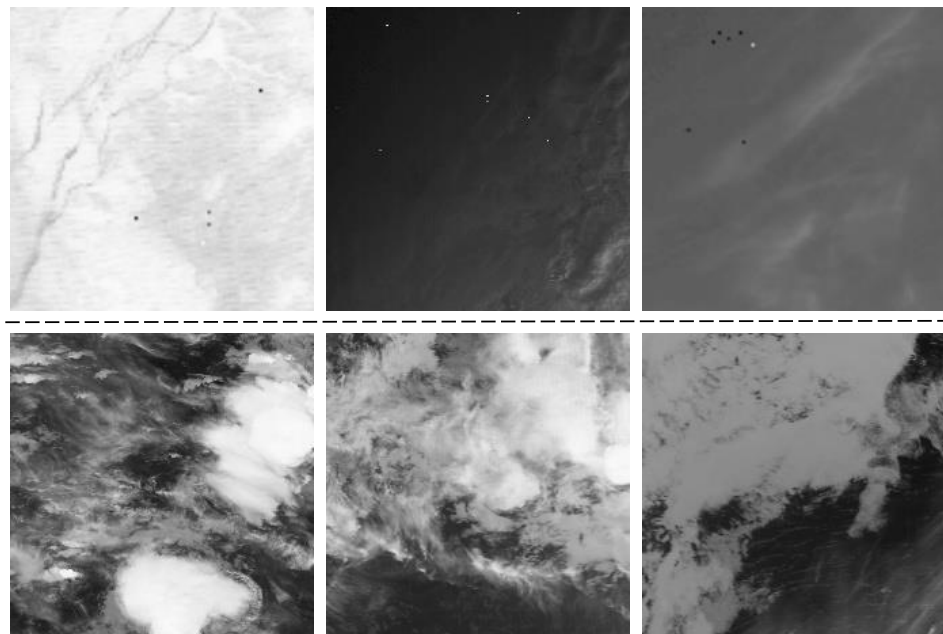


Figure 5. Some example images of a noise point dataset are shown.

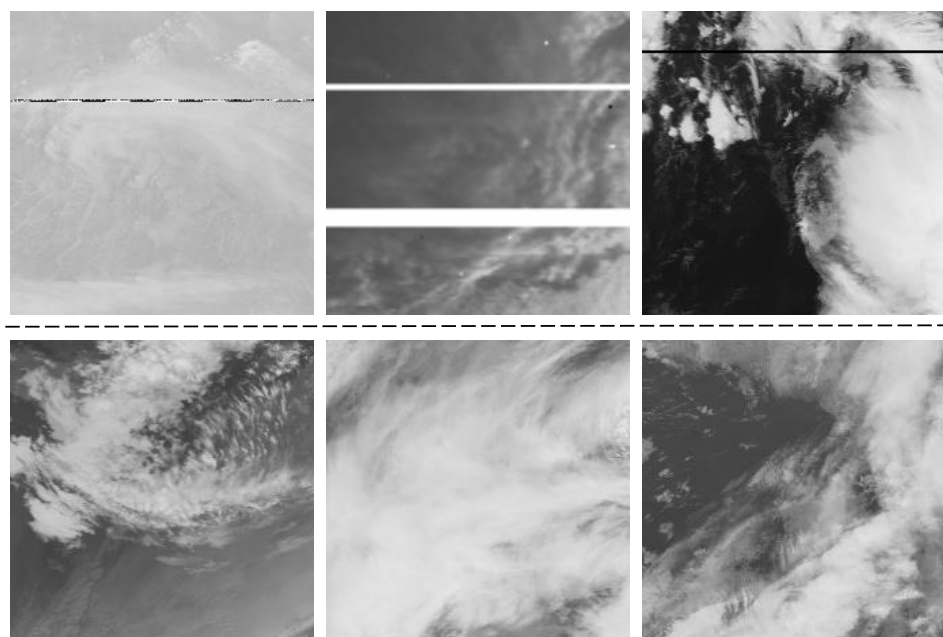


Figure 6. Some example images of a noise line dataset are displayed.

4.2. Experimental Setup

4.2.1. Implementation Details

Our proposed method is implemented using the PyTorch framework, a widely utilized deep learning library. The experimental evaluations are conducted on a high-performance workstation featuring an Intel(R) Core(TM) i9-10920X CPU clocked at 3.50 GHz. The computational power is enhanced by two GeForce RTX 3090 24GB TURBO GPUs, accelerating the training process significantly. To optimize our model, we employ stochastic gradient descent (SGD) as our chosen optimization algorithm, widely acknowledged in deep learning research. Experiments are conducted with a carefully chosen mini-batch size of 256 for a balance between computational efficiency and model convergence. The initial learning rate is set to $10^{-1.5}$ and follows a linear decay schedule, progressively decreasing with each epoch until reaching 10^{-6} . Additionally, cross-entropy was chosen as the loss function to ensure an effective evaluation of classification performance. This adaptive learning rate strategy ensures effective convergence and stability during the training process. The comprehensive evaluation involves 200 epochs, providing a rigorous exploration of the model's performance over an extended training duration to capture intricate patterns and trends in the learning dynamics, contributing to a thorough understanding of its capabilities and robustness.

Below is a brief introduction to the parameter settings for the comparative methods. In classification methods, the Logistic Regression method uses the 'lbfgs' solver with the regularization strength set to 1. The K-Nearest Neighbors method sets the number of neighbors to 5, uses Minkowski as the distance metric, and 'uniform' as the weight. The Decision Tree method sets the minimum number of samples for splitting to 2 and the minimum number of samples per leaf node to 1. The Random Forest method sets the number of trees to 100 and 'sqrt' as the maximum number of features. The Multilayer Perceptron method uses a continuous learning rate decay and 'relu' as the activation function. The AdaBoost method uses the default decision tree as the base learner, with the learning rate set to 1 and the number of trees set to 50. The Support Vector Machine method uses the 'rbf' kernel function, with the regularization parameter set to 1. For other deep learning classification algorithms, the parameters remain unchanged, such as the ResNet50 model, where the layers contain 3, 4, 6, and 3 basic blocks, respectively. For segmentation algorithms, the parameters are set to default values. For instance, the U-Net network defaults to using ResNet34 as the backbone network, sigmoid as the activation function, and ImageNet as the pretrained weights. The DeepLabV3 network uses ResNet50 as the backbone, with cross-entropy as the default loss function.

In the evaluation of noise image classification, we systematically compare the effectiveness of our proposed approach with a diverse range of ten classical and state-of-the-art techniques for classifying defects in satellite cloud images. This comprehensive analysis includes eight well-established traditional shallow methods: Logistic Regression [29], K-Nearest Neighbors [30], Naive Bayes Classification [31], Decision Tree [32], Random Forest [33], Multilayer Perceptron [34], AdaBoost [35], and Support Vector Machine [36]. Additionally, we incorporate two modern deep learning methods, namely, AlexNet [37] and ResNet50 [38]. In our assessment of noise region segmentation, we meticulously contrast various prominent image segmentation models, encompassing Unet [18], Unet++ [19], DeepLabV3 [20], DeepLabV3+ [21], A2-FPN [23], and MACU-Net [22]. We delve into a comprehensive comparison, examining their respective strengths, weaknesses, and performance across different noise types. This thorough analysis aims to provide insights into their applicability and effectiveness in addressing the challenges posed by noise region segmentation tasks.

4.2.2. Evaluation Protocol

Four evaluation metrics are commonly used to assess the performance of satellite cloud image defect classification methods and noise region segmentation methods: accuracy, precision, recall, F1 score, and mIoU. The definitions of these metrics are as follows:

- **Accuracy** is a measure that reflects the overall effectiveness of a classification model. It assesses the proportion of correctly predicted instances among all instances. A high accuracy score indicates that the model is making correct predictions across all classes, while a low accuracy suggests a higher rate of misclassifications. The definition of the accuracy metric is as follows:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}. \quad (5)$$

Accuracy gives an overall assessment of the model's ability to make correct predictions across all classes.

- **Precision** is a metric that focuses on the accuracy of positive predictions made by a model. It quantifies the model's ability to correctly identify instances belonging to the positive class. High precision indicates that when the model predicts a positive instance, it is likely to be correct, minimizing false positives. The definition of the precision metric is as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \quad (6)$$

TP and FP denote True Positives and False Positives. Precision is particularly important in situations where false positives carry significant consequences.

- **Recall** measures the effectiveness of a classification model in capturing all relevant instances of a specific class. It emphasizes the ability of the model to avoid missing positive instances, making it crucial in scenarios where false negatives (missing positives) have significant implications. A high recall score indicates a model that is sensitive to the presence of positive instances. The definition of the recall metric is as follows:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (7)$$

FN denotes False Negatives. Recall is important when the cost of missing positive instances (false negatives) is high, and it provides insight into how well the model identifies all relevant instances.

- **F1 Score** is a comprehensive metric that balances precision and recall. It is particularly useful in situations where there is an uneven distribution of classes or where there is a trade-off between false positives and false negatives. The F1 score is the harmonic mean of precision and recall, offering a single value that considers both the correctness of positive predictions and the model's ability to capture all relevant instances. The definition of F1 score is as follows:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (8)$$

The F1 score combines both precision and recall into a single metric, allowing for a comprehensive evaluation of a model's performance, especially in scenarios where there is a trade-off between false positives and false negatives.

- **mIoU**, i.e., mean Intersection over Union, is a widely used evaluation metric in semantic segmentation tasks. It assesses the accuracy of pixel-wise classification by measuring the overlap between predicted segmentation masks and ground truth

masks for each class. mIoU is the average IoU across all classes, where IoU is calculated as the ratio of the intersection area between the predicted and ground truth masks to their union area. Mathematically, IoU is expressed as follows:

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}. \quad (9)$$

TP denotes the number of true positive pixels (correctly classified pixels). FP is the number of false positive pixels (incorrectly classified pixels). FN is the number of false negative pixels (pixels missed by the prediction). mIoU is calculated by summing up the IoU of each class and dividing it by the total number of classes, which is defined as follows:

$$\text{mIoU} = \frac{1}{N} \sum_{i=1}^N \text{IoU}_i. \quad (10)$$

N is the total number of classes. In semantic segmentation, a higher mIoU value indicates better performance, meaning the model can accurately delineate different objects or regions within an image.

4.3. Experimental Results

We present a comprehensive quantitative comparison of our methodology with existing classification approaches using the noise point dataset, summarizing the results in Table 1. The table provides details on accuracy, precision, recall, and F1 score for all methods in the test dataset. Several key observations can be made from the outcomes presented in Table 1. First, deep learning methods outperform shallow methods, even when deep features are utilized. For example, compared to logistic regression, ResNet50 shows improvements of 5.60%, 7.18%, 11.20%, and 5.99% in accuracy, precision, recall, and F1 score, respectively. Shallow methods independently learn feature representations and classification models, limiting their discriminatory capacity for satellite cloud images containing noise points. In contrast, deep learning methods, which use convolutional neural networks for both feature representation and classification, consistently outperform shallow methods by enabling joint execution of noise point image recognition. Consequently, all deep methods consistently outperform shallow methods. Furthermore, our methodology demonstrates superior classification performance compared to other methods on noise point datasets. Compared to ResNet50, our approach achieves significant enhancements of 0.20%, 0.61%, and 0.21% in accuracy, recall, and F1 score, respectively. Compared to VGG16-AdvNet, our approach achieves significant enhancements of 6.8%, 2.08%, and 11.8% in accuracy, precision, and recall. This superiority is attributed to the adoption of transformer networks for classifying satellite cloud images, using self-attention mechanisms to capture long-range dependencies and global contextual information. This results in more effective feature learning across the entire image compared to convolutional networks. Additionally, transformers use various spatial hierarchies, which contribute to a more remarkable performance than convolutional networks. In summary, our method surpasses all baseline methods in noise point image recognition.

Figure 7 illustrates a selection of results from the noise point experiment. The first two images represent remote sensing satellite images with noise points, while the latter two are normal satellite images without noise. We compared the classification predictions of ResNet50, AlexNet, and our proposed model. In all four cases, our model achieved correct predictions, whereas ResNet50 only succeeded with the last image, and AlexNet misclassified all four images. Given the subtlety of the noise features in these images, these results highlight the robustness and superior performance of our model in detecting noise points.

Table 1. The accuracy, precision, recall, and F1 score of all methods are computed using satellite cloud images that include noise points.

Method	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
K-Nearest Neighbors [30]	84.00	98.99	68.70	81.11
Naive Bayes [31]	84.80	87.83	80.80	84.17
Decision Tree [32]	83.15	95.47	69.60	80.51
Random Forest [33]	82.55	98.37	66.20	79.14
AdaBoos [35]	88.55	97.30	79.30	87.38
Support Vector Machine [36]	90.95	99.88	82.00	90.06
Logistic Regression [29]	93.70	98.88	88.40	93.35
Multilayer Perceptron [34]	94.30	97.74	90.70	94.09
AlexNet [37]	97.00	98.16	95.80	96.96
ResNet50 [38]	98.95	99.59	98.30	98.94
MobileNetV2-Adv [39]	91.95	94.20	89.40	91.74
VGG16-AdvNet [40]	92.35	97.32	87.10	91.93
Ours	99.15	99.40	98.90	99.15

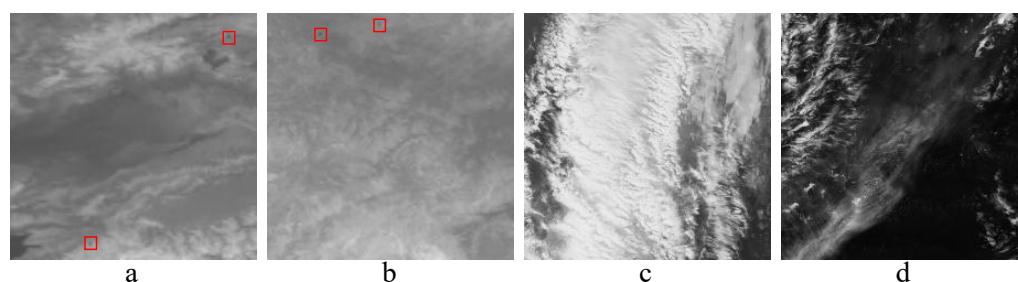


Figure 7. A selection of results from the noise points experiment. (a,b) are remote sensing satellite images containing noises, and (c,d) are normal remote sensing satellite images.

Furthermore, we conduct a comprehensive quantitative comparison of all methods in the noise line dataset, presenting the results in Table 2, including the accuracy, precision, recall, and F1 score metrics in the test dataset. From these findings, several observations emerge. First, deep learning methods outperform shallow methods when applied to noise lines, even when deep features are utilized. For example, compared to logistic regression, ResNet50 shows improvements of 0.71%, 1.75%, and 0.73% in accuracy, recall, and F1 score. Moreover, the improvement in these methods on noise lines is comparatively lower than that on noise points, since the distinct characteristics of noise lines make it easier for shallow methods to achieve better performance. Second, our method applied to noise lines exhibits superior classification performance compared to other methods. For example, relative to ResNet50, our approach achieves significant improvements of 0.20%, 0.41%, and 0.20% in accuracy, recall, and F1 score, respectively. Compared to MobileNetV2-Adv, our approach achieves significant enhancements of 4.05%, 8.2%, and 4.27% in accuracy, recall, and F1 score. This superiority is attributed to the utilization of transformer networks, which allows more effective feature learning across the entire image than convolutional networks, especially in capturing long-range dependencies and global contextual information through self-attention mechanisms.

Table 2. The accuracy, precision, recall, and F1 score of all methods are computed using satellite cloud images that include noise lines.

Method	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
K-Nearest Neighbors [30]	94.35	99.89	88.80	94.02
Naive Bayes [31]	90.65	96.89	84.00	89.98
Decision Tree [32]	94.95	99.13	90.70	94.73
Random Forest [33]	94.70	99.67	89.70	94.42
AdaBoost [35]	96.35	99.36	93.30	96.24
Support Vector Machine [36]	97.60	99.79	95.40	97.55
Logistic Regression [29]	98.35	99.69	97.00	98.33
Multilayer Perceptron [34]	98.25	99.49	97.00	98.23
AlexNet [37]	98.65	99.80	97.50	98.63
ResNet50 [38]	99.05	99.40	98.70	99.05
MobileNetV2-Adv [39]	95.20	99.45	90.90	94.98
VGG16-AdvNet [40]	98.10	99.90	96.30	98.07
Ours	99.25	99.40	99.10	99.25

Additionally, the improvement in our method compared to other methods for noise lines is lower than for noise points, as the distinctiveness of noise lines allows for easier identification. As shown in Table 3, while the precision of our classification method is slightly lower than that of the best-performing method (though the difference is minimal), its recall significantly outperforms the latter. Our approach employs a two-stage framework. The first stage selects images that contain noise, while the second stage identifies the specific locations of noise points and lines within these images. Existing methods often suffer from low recall when precision is high in the first stage, resulting in the omission of many noisy images. In contrast, our method achieves a better balance between precision and recall, ensuring more comprehensive detection of noise. For example, if a pixel is misclassified as noise during the detection stage, the segmentation stage will not segment it, thereby preventing the error that could arise from instability in a single model. This two-stage design effectively mitigates the impact of minor accuracy losses on the overall performance. Moreover, in the field of meteorological image processing, noisy images can significantly affect subsequent tasks. Therefore, the model must aim to recall all potential noisy images. Our approach emphasizes identifying all noisy images, while the two-stage algorithm helps prevent errors caused by insufficient accuracy, resulting in better defect recognition performance. In summary, our method outperforms all comparison methods in noise line image recognition.

Figure 8 showcases a selection of results from the noise line experiment. The first two images represent remote sensing satellite images with noise lines, while the latter two are normal satellite images without noise lines. We compared the classification predictions of ResNet50, AlexNet, and our proposed model. Among these four images, our model achieved correct predictions across all cases. In contrast, AlexNet correctly classified only the first two images containing noise lines but misclassified the last two, while ResNet50 failed to correctly predict any of the images. These results highlight the superior performance of our model in detecting noise lines. Particularly for noise lines, which exhibit relatively continuous features and are more challenging to discern, our model demonstrates remarkable stability and precision. This performance advantage stems from the ability of our model to leverage self-attention mechanisms, capturing global contextual information and extracting key features associated with noise lines more effectively. In comparison, the predictions of AlexNet and ResNet50 exhibit significant inconsistencies, underscoring their limitations in handling such tasks.

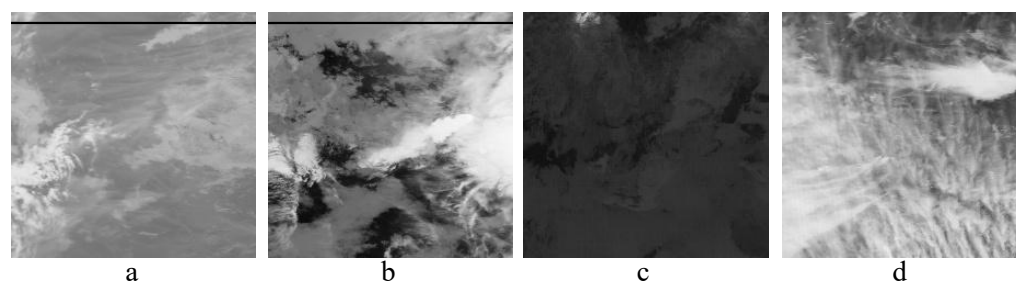


Figure 8. A selection of results from the noise lines experiment. (a,b) are remote sensing satellite images containing lines, and (c,d) are normal remote sensing satellite images.

We also experimented with replacing the DeiT model with the ViT [41] model. The results are shown in Table 3. The results showed that the ViT model performed similarly to the DeiT model, and even slightly inferior to DeiT model across various accuracy metrics. However, the parameter count and Flops of the ViT model are 4–5 times that of the DeiT model. From the perspective of parameter scale and inference time, the ViT model and other more complex Transformer models often fail to meet the efficiency requirements of real-world business applications. Although these models may offer high performance in certain scenarios, their high computational cost and extended inference time make them less practical for deployment. Therefore, considering both performance and efficiency, we adopted DeiT as the backbone to identify images containing noise. The DeiT model not only delivers excellent accuracy but also provides faster inference speeds in resource-constrained environments, making it a practical solution for real-world business needs.

Table 3. The accuracy, precision, recall, and F1 score of the DeiT model and ViT model.

Method	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
ViT (points)	98.10	98.59	97.60	98.10
DeiT (points)	99.15	99.40	98.90	99.15
ViT (lines)	99.00	99.60	98.40	99.00
DeiT (lines)	99.25	99.40	99.10	99.25

Additionally, we performed Flops and parameter analyses on the classification methods used, with the results shown in Table 4. In this discussion, we focus exclusively on deep learning models, excluding traditional machine learning models. From the table, it is evident that our model has Flops and parameter sizes close to the minimum, while achieving the shortest inference time. Our model successfully achieves lightweight design while maintaining high accuracy and stability, making it highly suitable for resource-constrained environments. Furthermore, the reduced model complexity enhances its scalability and deployment flexibility in practical applications, offering a robust solution for real-time processing demands.

Table 4. Flops and parameters of various classification methods, as well as the test time of running noise points and noise lines experiment.

Method	Flops	Parameters	Noise Points (s)	Noise Lines (s)
AlexNet [37]	7.10×10^8	6.11×10^7	0.264	0.245
ResNet50 [38]	4.10×10^9	2.56×10^7	1.873	1.970
MobileNetV2-Adv [39]	3.06×10^8	1.31×10^7	1.375	1.356
VGG16-AdvNet [40]	1.55×10^{10}	1.34×10^8	0.505	0.514
Ours	4.63×10^9	2.17×10^7	0.501	0.505

4.4. Ablation Study

To validate the importance of self-attention in our model, we conducted ablation studies. Self-attention is the core component of the transformer architecture; when removing the self-attention layers, our DeiT-based network essentially degrades to a simple Multilayer Perceptron (MLP). Our experimental results demonstrate that removing self-attention leads to significant performance degradation. The experimental results of ablation are shown in Tables 5 and 6 below.

Table 5. The results of self-attention module ablation using satellite cloud images that include noise points.

Method	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
without self-attention	94.30	97.74	90.70	94.09
with self-attention	99.15	99.40	98.90	99.15

Table 6. The results of self-attention module ablation using satellite cloud images that include noise lines.

Method	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
without self-attention	98.25	99.49	97.00	98.23
with self-attention	99.25	99.40	99.10	99.25

Specifically, for the noise line detection task, the complete DeiT architecture with self-attention achieves 99.25% accuracy, 99.40% precision, 99.10% recall, and 99.25% F1-score. In contrast, removing self-attention results in lower performance with 98.25% accuracy, 99.49% precision, 97.00% recall, and 98.23% F1-score. For the noise point detection task, the performance gap is even more pronounced. With self-attention, the model achieves 99.15% accuracy, 99.40% precision, 98.90% recall, and 99.15% F1-score, while without self-attention, these metrics drop to 94.30% accuracy, 97.74% precision, 90.70% recall, and 94.09% F1-score. These results confirm the crucial role of self-attention in capturing both global and local dependencies of noise patterns, particularly for the more challenging noise point detection task.

4.5. Performance on Different Subclasses

In this study, our primary objective was to evaluate the performance of various methods across two distinct classification tasks: noise point classification and noise line classification. We conducted a comprehensive analysis, utilizing bar charts to assess the performance of each method in every category. Figures 9–11 illustrate the precision, recall, and F1 metric indicators for all methods in noise point and normal images, respectively.

The results shown in Figure 9 indicate that all methods achieved a higher level of precision when dealing with images containing noise points compared to normal images. On the other hand, Figure 10 reveals that all methods achieved a higher level of recall when dealing with normal images. These findings suggest that, in the task of classifying noise points, most methods focused on evaluating the performance of classifiers in accurately identifying and classifying instances that were affected by noise points within the dataset.

To achieve a balanced representation of the two metrics, we have presented the F1 score for all methods in Figure 11, which effectively provides an average assessment of the impact of precision and recall evaluation metrics on the model. These figures indicate that our method successfully balances the precision and recall metrics, resulting in high performance on both indicators. Additionally, our method consistently achieves high F1 scores for both noise points and normal images. In contrast, shallow methods such as

KNN often struggle to perform well in both categories simultaneously. In conclusion, our method demonstrates superior overall performance in all categories when compared to other methods.

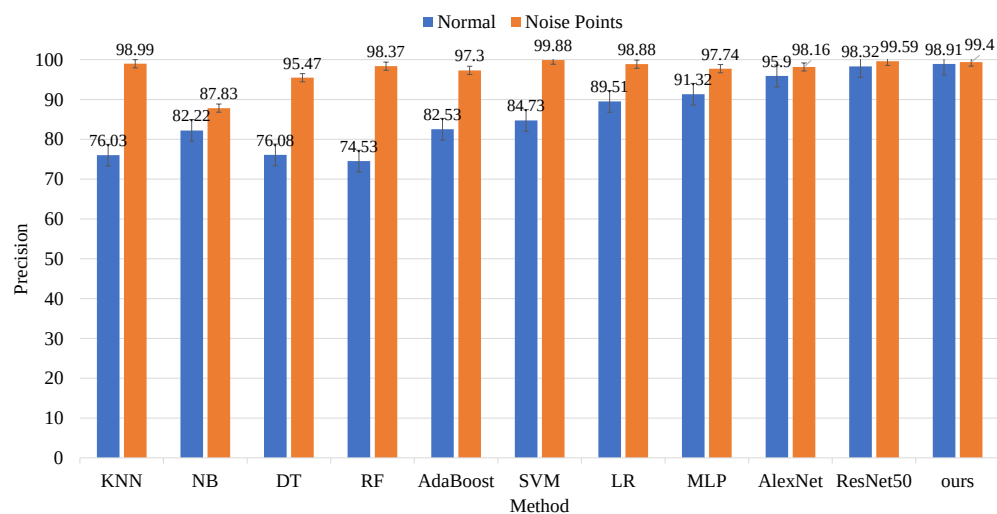


Figure 9. The precision of all methods on normal and noise point images.

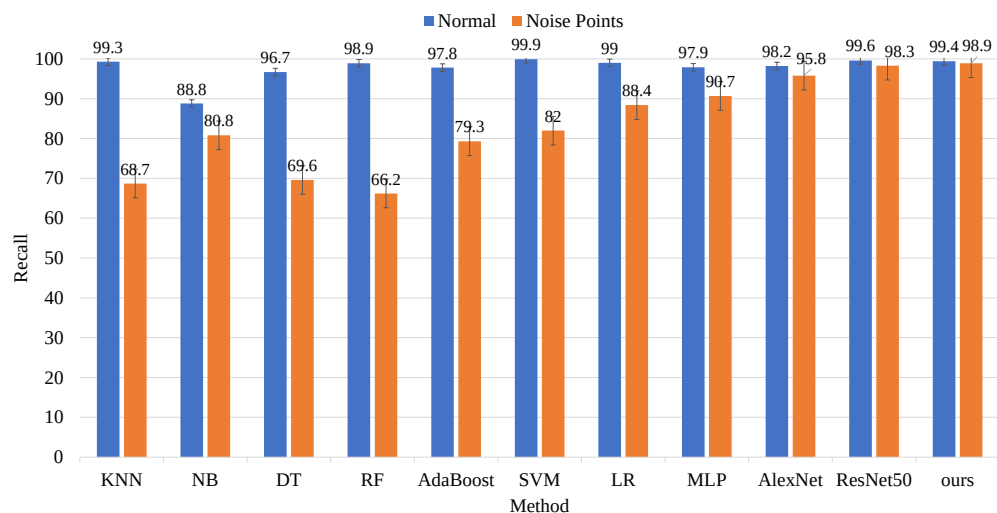


Figure 10. The recall of all methods on normal and noise point images.

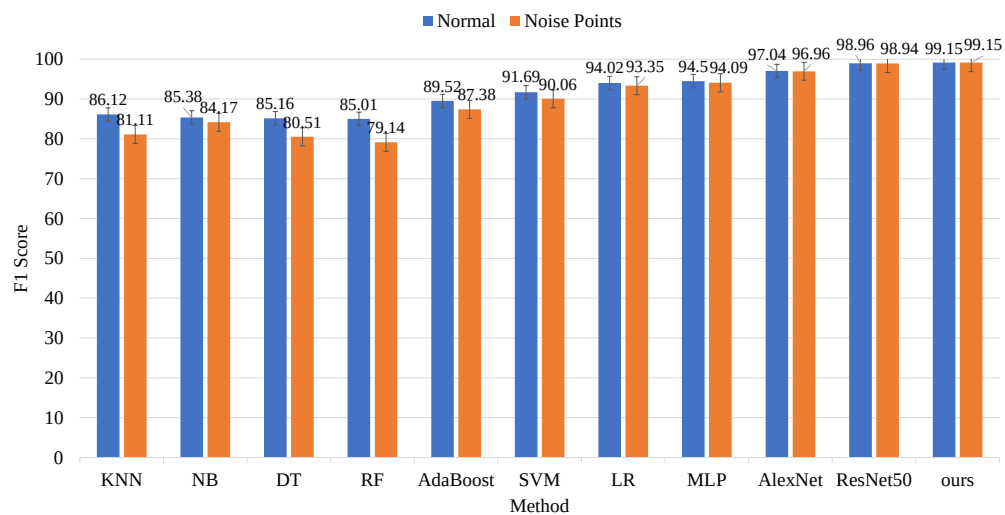


Figure 11. The F1 score for all methods on normal and noise point images.

The performance of various methods in classifying noise line images was further analyzed. The precision, recall, and F1 scores were evaluated for all methods on normal and noise line images, as shown in Figures 12–14, respectively. Based on these figures, three observations can be made. First, most shallow methods exhibit a gap in precision between different categories. For example, the KNN method achieves a precision of 9.97% higher in noise line images compared to normal images. In contrast, the ResNet50 method achieves similar precision values of 98.71% and 99.39% in the two categories, respectively, indicating the ability of deep learning methods to learn feature representations and classification models end-to-end, resulting in better precision. Second, most methods perform well in recall metrics, indicating their effectiveness in recalling satellite cloud images with noise line features. Finally, our method achieves the highest F1 score values in both normal and noise line images. This can be attributed to the fact that noise lines typically cover larger areas, and our method effectively utilizes the transformer network to learn long-range features, enabling an effective distinction between normal and noise line images. In conclusion, our method outperforms all other methods in classifying noise line images.

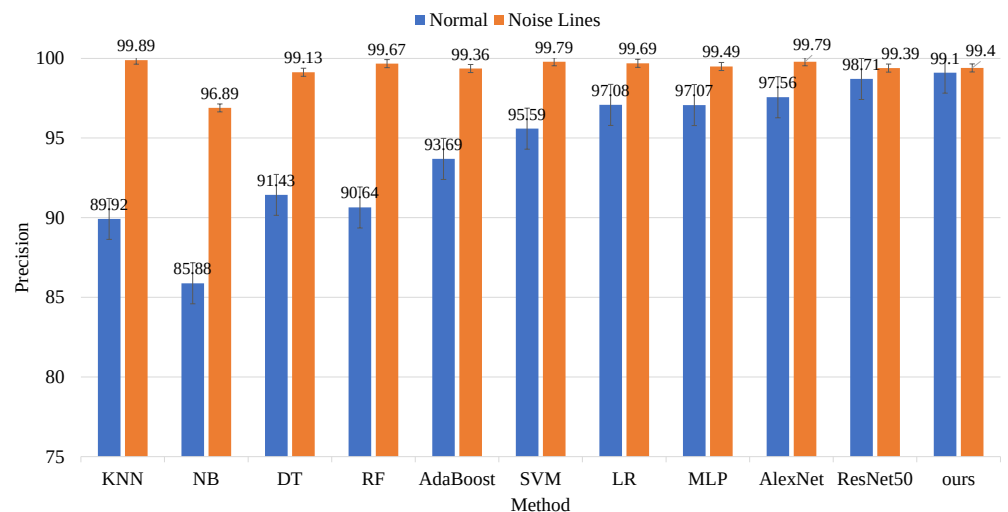


Figure 12. The precision of all methods on normal and noise line images.

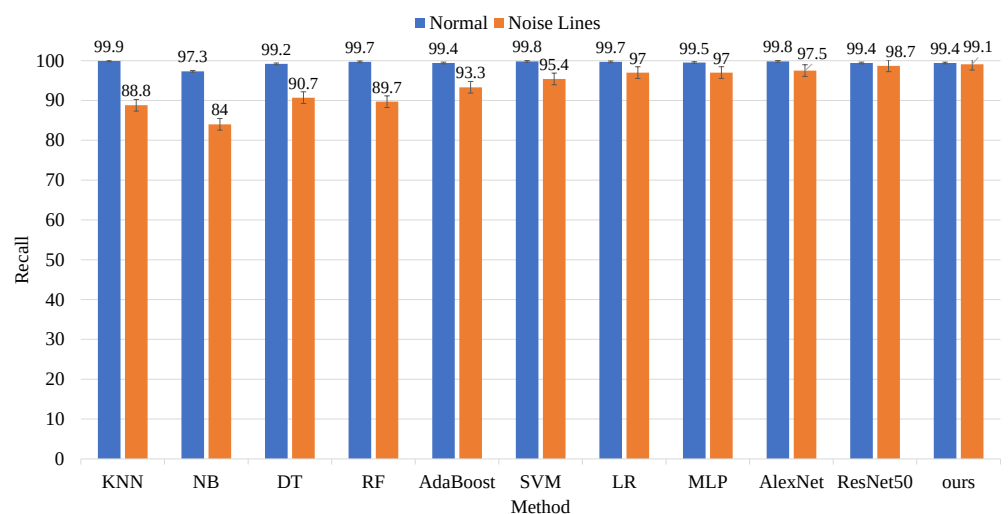


Figure 13. The recall of all methods on normal and noise line images.

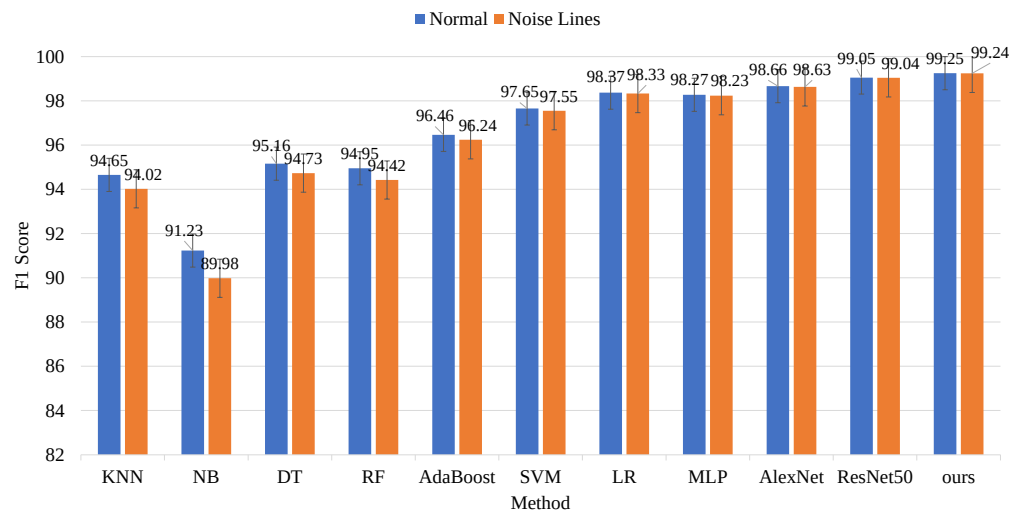


Figure 14. The F1 score for all methods on normal and noise line images.

4.6. Comparison Results of Noise Region Segmentations

In this paper, we employ a two-level deep defect recognition framework to identify noise points and lines in meteorological satellite images. The framework begins with the application of a transformer-based method for image classification. Subsequently, we utilize a pseudo-label-based training strategy and popular image segmentation models (e.g., Unet) to detect the regions containing noise points and lines. By combining image classification and image segmentation approaches, we achieve a balance between processing efficiency and performance. While image classification methods typically filter normal images and require less inference time, image segmentation models only need to deal with the noise images, resulting in time and resource savings. The performance comparison of popular methods in identifying noise points and lines in images is shown in Table 7.

Table 7. Performance comparison of noise region segmentation methods using mIoU.

Type	Unet	Unet++	DeepLab V3	DeepLab V3+	A2-FPN	MACU-Net
Noise Points	58.50	58.12	52.24	57.05	58.06	57.09
Noise Lines	69.80	72.14	66.42	71.47	71.48	71.99

Table 7 shows the performance of various methods in segmenting noise points. We observe that DeepLabV3 and DeepLabV3+ exhibit poorer performance compared to other methods. This is mainly due to their reliance on atrous convolutions for contextual information extraction from satellite images. However, the small size of noise points poses a challenge for these methods, as their networks struggle to effectively capture features of such minuscule points, leading to inaccurate detection. Conversely, encoder–decoder-based methods, such as Unet, demonstrate higher accuracy in segmenting noise regions. For example, Unet++ achieves an 11.26% improvement over DeepLabV3, attributed to the precise feature reconstruction capability of the encoder–decoder architecture, which enhances the model’s ability to delineate small objects accurately. Moreover, the U-shaped architecture commonly adopted by encoder–decoder models incorporates skip connections to preserve fine details and effectively localize small structures, making it highly effective in detecting noise points in satellite images. Similarly, for noise lines, as presented in Table 7, we observe that all methods achieve higher accuracy compared to noise points. This is because noise lines typically have a larger size than noise points, enabling image segmentation

models to effectively detect them. Overall, our pseudo-label-based training strategy enables the development of effective noise region segmentation models for accurately detecting noise pixels.

We present the segmentation results of various methods in Figures 15 and 16. In Figure 15, comprising six columns and six rows, the left three columns display meteorological satellite images, ground truth, and predicted results, respectively. The right three columns are similar to the left set. Notably, we observe that these methods may overlook small noise points due to their challenging detection. Additionally, the accuracy of the right images is lower than that of the left images. This disparity arises because background elements, such as clouds, in the right images can interfere with the model's ability to detect noise points effectively.

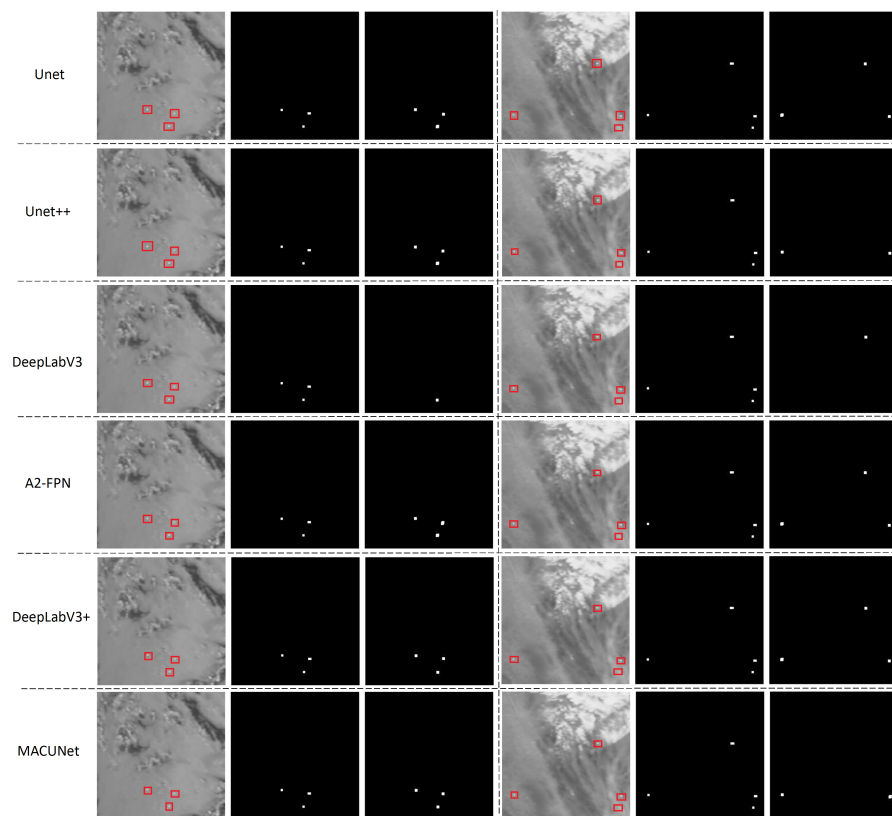


Figure 15. The visualization results of image segmentation methods for meteorological satellite images containing noise points. The noise points have been marked with red boxes.

Figure 16 illustrates the visualization of certain methods for detecting noise lines. The left three columns exhibit satellite images, ground truth, and predicted results, while the right columns are similar to the left set. We notice that trained models can identify various types of noise lines, even those with complex structures. Although our created dataset predominantly comprises black or white noise, the model effectively discerns these intricate noise lines. Unlike noise points, detecting noise lines may only capture a portion of the area, leaving gaps in the results. Hence, future research could focus on designing post-processing methods to fully capture noise lines based on their horizontal or vertical characteristics. In summary, these visualizations affirm the efficacy of our training strategy in developing a robust model for segmenting noise regions and identifying defective pixels in satellite images.

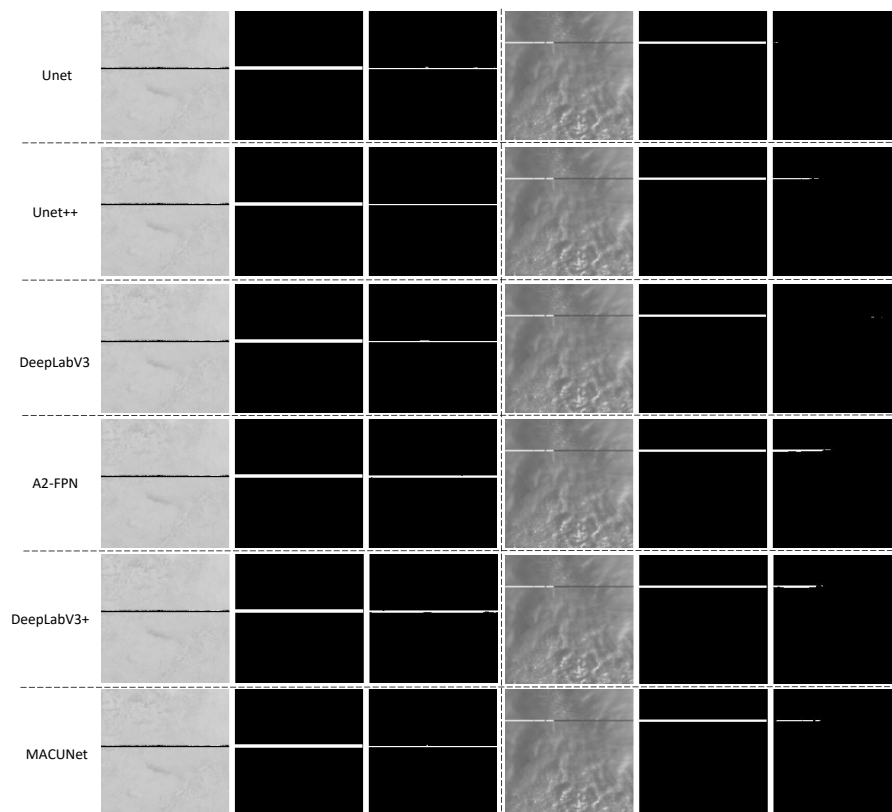


Figure 16. The visualization results of image segmentation methods for meteorological satellite images containing noise lines.

Additionally, we conducted an analysis of the Flops and parameters for the semantic segmentation methods used, with the results presented in Table 8. It can be observed that the model with the largest number of parameters, Unet++, has a parameter count in the order of 10^7 , while the model with the highest Flops, Unet, also has Flops in the order of 10^{10} . The models listed in the table were able to complete inference on large datasets within 10 s, indicating high efficiency and good accuracy.

Table 8. Flops and parameters of various semantic segmentation methods, as well as the test time of running noise points and noise lines experiments.

Type	Unet	Unet++	DeepLab V3	DeepLab V3+	A2-FPN	MACU-Net
Flops	2.27×10^{10}	1.23×10^{10}	1.28×10^{10}	3.51×10^9	2.59×10^9	5.64×10^9
Parameters	1.48×10^7	1.60×10^7	1.59×10^7	1.23×10^7	1.22×10^7	5.15×10^6
Points(s)	8.66	6.39	7.46	4.19	3.93	8.48
Lines(s)	9.25	6.28	7.39	4.04	3.84	8.40

5. Conclusions

In this paper, we study the problem of recognizing defects in meteorological satellite images. This problem has not been explored previously using deep learning methods. We propose a two-level deep defect recognition framework for meteorological satellite images. In the first level, we introduce a transformer-based noise image classification module to identify whether a meteorological satellite image contains noise points or noise lines. Additionally, we collect and release two satellite cloud image datasets for evaluating our proposed method. In the second level, we present a pseudo-label-based training strategy for constructing image segmentation models, enabling further detection of regions containing

noise points and lines within meteorological satellite images. We compare popular image segmentation models using the proposed training strategy on real noise point or noise line images. We conduct comprehensive experiments to assess the performance of our proposed framework method in addressing the satellite cloud image defect recognition problem. The results demonstrate the effectiveness of our framework in identifying noise types and accurately detecting noise regions in meteorological satellite images.

However, our method has certain limitations. The primary constraint of the current DeepDR framework lies in its validation being predominantly conducted on meteorological satellite imagery, while its effectiveness on other types of satellite imagery remains to be explored. Different satellite systems capture distinct spectral bands and possess varying imaging characteristics, which may affect the generalization capability of our method. In future work, we aim to enhance DeepDR to develop a more versatile framework capable of accommodating multi-modal remote sensing data, thereby expanding its applicability and value across broader remote sensing domains.

Author Contributions: Conceptualization, X.Z. and M.L.; methodology, X.Z.; software, X.Z. and M.L.; validation, X.Z. and C.F.; formal analysis, C.F.; investigation, X.Z.; resources, X.C.; data curation, X.C. and L.W.; writing—original draft preparation, X.Z.; writing—review and editing, C.F., X.C. and Y.Y.; visualization, L.W. and X.C.; supervision, Y.Y.; project administration, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Key Technology Research and Development Program of China (Grant Nos. 2023YFB3905300 and 2023YFB3905302).

Data Availability Statement: The dataset have been made available in this link: <https://github.com/weather-tech/DeepDR.git> (accessed on 26 January 2025).

Conflicts of Interest: Auhtors declare no conflicts of interests.

References

- Kim, M.; Song, H.; Kim, Y. Direct Short-Term Forecast of Photovoltaic Power through a Comparative Study between COMS and Himawari-8 Meteorological Satellite Images in a Deep Neural Network. *Remote. Sens.* **2020**, *12*, 2357. [\[CrossRef\]](#)
- Vyas, S.S.; Bhattacharya, B.K. Agricultural drought early warning from geostationary meteorological satellites: Concept and demonstration over semi-arid tract in India. *Environ. Monit. Assess.* **2020**, *192*, 1–15. [\[CrossRef\]](#) [\[PubMed\]](#)
- Wang, Z.; Zhang, D. Progressive switching median filter for the removal of impulse noise from highly corrupted images. *IEEE Trans. Circuits Syst. II Analog. Digit. Signal Process.* **1999**, *46*, 78–80. [\[CrossRef\]](#) [\[PubMed\]](#)
- Chen, T.; Wu, H.R. Adaptive impulse detection using center-weighted median filters. *IEEE Signal Process. Lett.* **2001**, *8*, 1–3. [\[CrossRef\]](#)
- Chen, T.; Ma, K.K.; Chen, L.H. Tri-state median filter for image denoising. *IEEE Trans. Image Process.* **1999**, *8*, 1834–1838. [\[CrossRef\]](#) [\[PubMed\]](#)
- Ferdous, H.; Siraj, T.; Setu, S.J.; Anwar, M.M.; Rahman, M.A. Machine learning approach towards satellite image classification. In *Proceedings of the International Conference on Trends in Computational and Cognitive Engineering: Proceedings of TCCE 2020*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 627–637.
- Valero Medina, J.A.; Alzate Atehortúa, B.E. Comparison of maximum likelihood, support vector machines, and random forest techniques in satellite images classification. *Tecnura* **2019**, *23*, 13–26. [\[CrossRef\]](#)
- Kulkarni, S.; Kelkar, V. Classification of multispectral satellite images using ensemble techniques of bagging, boosting and adaboost. In *Proceedings of the 2014 International Conference on Circuits, Systems, Communication and Information Technology Applications (CSCITA)*, Mumbai, India, 4–5 April 2014; IEEE: New York, NY, USA, 2014; pp. 253–258.
- Unnikrishnan, A.; Sowmya, V.; Soman, K. Deep AlexNet with reduced number of trainable parameters for satellite image classification. *Procedia Comput. Sci.* **2018**, *143*, 931–938. [\[CrossRef\]](#)
- Pritt, M.; Chern, G. Satellite image classification with deep learning. In *Proceedings of the 2017 IEEE applied imagery pattern recognition workshop (AIPR)*, Washington, DC, USA, 10–17 October 2017; IEEE: New York, NY, USA, 2017; pp. 1–7.
- Wang, D.; Zhuang, L.; Gao, L.; Sun, X.; Zhao, X.; Plaza, A. Sliding dual-window-inspired reconstruction network for hyperspectral anomaly detection. *IEEE Trans. Geosci. Remote. Sens.* **2024**, *62*, 1–15. [\[CrossRef\]](#)

12. Wang, D.; Zhuang, L.; Gao, L.; Sun, X.; Huang, M.; Plaza, A. BockNet: Blind-block reconstruction network with a guard window for hyperspectral anomaly detection. *IEEE Trans. Geosci. Remote. Sens.* **2023**, *61*, 1–16. [\[CrossRef\]](#)
13. Li, Y.; Jiang, T.; Xie, W.; Lei, J.; Du, Q. Sparse coding-inspired GAN for hyperspectral anomaly detection in weakly supervised learning. *IEEE Trans. Geosci. Remote. Sens.* **2021**, *60*, 1–11. [\[CrossRef\]](#)
14. Cheng, X.; Zhang, M.; Lin, S.; Li, Y.; Wang, H. Deep self-representation learning framework for hyperspectral anomaly detection. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 1–16. [\[CrossRef\]](#)
15. Zhuang, L.; Ng, M.K.; Gao, L.; Wang, Z. Eigen-CNN: Eigenimages Plus Eigennoise Level Maps Guided Network for Hyperspectral Image Denoising. *IEEE Trans. Geosci. Remote. Sens.* **2024**, *62*, 1–18. [\[CrossRef\]](#)
16. Wang, M.; Gao, L.; Ren, L.; Sun, X.; Chanussot, J. Hyperspectral Simultaneous Anomaly Detection and Denoising: Insights From Integrative Perspective. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* **2024**, *17*, 13966–13980. [\[CrossRef\]](#)
17. Li, X.; Ding, M.; Gu, Y.; Pižurica, A. An end-to-end framework for joint denoising and classification of hyperspectral images. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 3269–3283. [\[CrossRef\]](#) [\[PubMed\]](#)
18. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015; proceedings, part III 18; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
19. Zhou, Z.; Rahman Siddiquee, M.M.; Tajbakhsh, N.; Liang, J. Unet++: A nested u-net architecture for medical image segmentation. In Proceedings of the Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, 20 September 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 3–11.
20. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv* **2017**, arXiv:1706.05587.
21. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818.
22. Li, R.; Duan, C.; Zheng, S.; Zhang, C.; Atkinson, P.M. MACU-Net for Semantic Segmentation of Fine-Resolution Remotely Sensed Images. *IEEE Geosci. Remote. Sens. Lett.* **2022**, *19*, 1–5. [\[CrossRef\]](#)
23. Li, R.; Wang, L.; Zhang, C.; Duan, C.; Zheng, S. A2-FPN for semantic segmentation of fine-resolution remotely sensed images. *Int. J. Remote. Sens.* **2022**, *43*, 1131–1155. [\[CrossRef\]](#)
24. Ma, Y.; Wang, Y.; Liu, X.; Wang, H. SWINT-RESNet: An improved remote sensing image segmentation model based on Transformer. *IEEE Geosci. Remote. Sens. Lett.* **2024**, *21*, 8003005. [\[CrossRef\]](#)
25. Li, J.; Cheng, S. AFENet: An Attention-Focused Feature Enhancement Network for the Efficient Semantic Segmentation of Remote Sensing Images. *Remote. Sens.* **2024**, *16*, 4392. [\[CrossRef\]](#)
26. Kang, X.; Hong, Y.; Duan, P.; Li, S. Fusion of hierarchical class graphs for remote sensing semantic segmentation. *Inf. Fusion* **2024**, *109*, 102409. [\[CrossRef\]](#)
27. Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; Jégou, H. Training data-efficient image transformers & distillation through attention. In Proceedings of the International Conference on Machine Learning. PMLR, Online, 18–24 July 2021; pp. 10347–10357.
28. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–11.
29. Hosmer, D.W., Jr.; Lemeshow, S.; Sturdivant, R.X. *Applied Logistic Regression*; John Wiley & Sons: Hoboken, NJ, USA, 2013; Volume 398.
30. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [\[CrossRef\]](#)
31. Zhang, H. The optimality of naive Bayes. *Aa* **2004**, *1*, 3.
32. Quinlan, J.R. Induction of decision trees. *Mach. Learn.* **1986**, *1*, 81–106. [\[CrossRef\]](#)
33. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [\[CrossRef\]](#)
34. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [\[CrossRef\]](#)
35. Freund, Y.; Schapire, R.; Abe, N. A short introduction to boosting. *J.-Jpn. Soc. Artif. Intell.* **1999**, *14*, 1612.
36. Hearst, M.A.; Dumais, S.T.; Osuna, E.; Platt, J.; Scholkopf, B. Support vector machines. *IEEE Intell. Syst. Their Appl.* **1998**, *13*, 18–28. [\[CrossRef\]](#)
37. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1–9. [\[CrossRef\]](#)
38. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.

39. Cao, L. A MobileNetV2 model of transfer learning is employed for remote sensing image classification. *Adv. Eng. Technol. Res.* **2024**, *10*, 596. [[CrossRef](#)]
40. Xie, M.; Tang, Q.; Yang, K.; Ma, Y.; Zhao, S.; Feng, X.; Hao, W. Image classification based on improved VGG network. In Proceedings of the Fifth International Conference on Computer Vision and Data Mining (ICCVDM), Changchun, China, 19–21 July 2024; Volume 13272, pp. 352–356.
41. Dosovitskiy, A. An image is worth 16×16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.