Vol. 06, No. SI*, pp. 201 –221 (2025)
ISSN: 2708-0757

JOURNAL OF APPLIED SCIENCE AND TECHNOLOGY TRENDS

www.jastt.org

HSA-CNN: A Hybrid Spectral-Attention Multi-Agent Framework for Explainable Cloud Detection in Multispectral Remote Sensing Imagery

K.P. Swain[1], S.K. Mohapatra[2], Soumya Ranjan Nayak[3], Ashish Singh[3*]

[1]*Department of ETC, Trident Academy of Technology, Bhubaneswar, Odisha,* kaleep.swain@gmail.com
[2]*Department of Computer Science and Engineering (AI&ML), Dayananda Sagar University, School of Engineering, Bengaluru, Karnataka, India,* sumantkumar-aiml@dsu.edu.in
[3]*School of Computer Engineering, KIIT Deemed To Be University, Bhubaneswar, Odisha,* nayak.soumya17@gmail.com, ashishashish307@gmail.com

**Correspondence:* ashishashish307@gmail.com

Abstract

Cloud detection is an essential preprocessing step in a remote sensing application. However, the presence of clouds and their shadows severely undermines the accuracy of surface observations and subsequent analysis. Reliable identification of thin clouds and cloud shadows remains a problem to which even state-of-the-art deep learning-based cloud detection methods have not provided a solution, due to spectral ambiguity, spatial variability, and the lack of model uncertainty awareness. This work presents HSA-CNN, a hybrid spectral, attention, multi-agent deep learning framework that accurately and explainably identifies pixel-wise clouds in multispectral satellite imagery. The proposed architecture is a U-Net-based encoder-decoder architecture complemented by Spectral Directional Kernel (SDK) blocks for multi-scale feature extraction and integrating a set of specialized agents, including a transformer-based spectral attention agent, a MobileNet-based spatial context agent, a bidirectional LSTM-based temporal sequence agent, and a Bayesian uncertainty agent. This meta-agent orchestration mechanism performs confidence-aware, per-pixel expert selection and ensemble fusion, enabling robust predictions and reliable uncertainty estimation. The experimental results show that HSA and CNN can accurately classify four cloud categories: clear sky, thick cloud, thin cloud, and cloud shadow. Moreover, it significantly improves thin cloud discrimination and prediction stability. Furthermore, the framework can provide interpretable outputs via attention maps, agent-weight visualizations, and pixel-level uncertainty maps, which improve transparency and operational trust. The proposed method is a powerful, interpretable tool in remote sensing that can be used for atmospheric correction, environmental monitoring, and climate analysis.

Keywords: *Cloud Detection; Multispectral Remote Sensing; Deep Learning; Multi-Agent Systems; Spectral Attention; Explainable Artificial Intelligence*

Received: November 12th, 2025 / Revised: December 21st, 2025 /Accepted: December 27th, 2025 / Online: December 31st, 2025

I. INTRODUCTION

Cloud detection is a vital component of processing optical remote sensing data. This is because the clouds and their shadows hide the surface, making it difficult to use the data for subsequent tasks (e.g., land cover classification, atmospheric correction, environmental monitoring, and climate analysis) with high accuracy. Inaccurate cloud masking may result in mistakes in the entire Earth observation pipeline. Therefore, dependable cloud detection is essential for earth observation systems designed for operational use. Conventional cloud detection techniques mainly use spectral thresholding, physical models, and handcrafted indices. These methods are computationally efficient but often fail under complex atmospheric conditions and have limited generalization capabilities across sensors and geographic regions [1].

Due to the rapid deep learning progress, convolutional neural networks (CNNs) represent the most popular method for cloud detection in multispectral satellite imagery. Pixel-wise segmentation in encoder-decoder architectures such as U-Net and its variants has been significantly improved by directly learning hierarchical spectral-spatial representations from data [2]. Moreover, multi-scale feature fusion and attention mechanisms that enable the model to focus on relevant spectral bands and spatial regions in cloud formations have led to further improvements [3]. However, existing deep learning models still

doi:10.38094/jastt605741

struggle to differentiate thin clouds and cloud shadows that spectrally resemble land or water surfaces [4].

Moreover, a limitation of current cloud detection methods is most concerned with uncertainty awareness and explainability. The majority of deep learning models make deterministic predictions and do not provide confidence levels, which makes their trustworthiness in safety, critical, and large-scale scenarios questionable. Inter alia, many of the cutting-edge approaches are constructed as monolithic architectures, it is arduous to comprehend their decision-making process or modify them to other sensor settings [5]. These problems call for the development of cloud detection systems that are not only accurate but also stable, interpretable, and capable of uncertainty estimation.

Addressing the above constraints, this paper proposes HSA-CNN, a hybrid spectral, attention, multi-agent deep learning framework for explainable pixel-wise cloud detection in multispectral remote sensing imagery. The suggested method implements a multi-agent learning paradigm where agents specializing in different areas independently model spectral dependencies, spatial context, temporal patterns, and predictive uncertainty. A confident, aware meta-agent coordinator that pixel-wise merges the agents' outputs, not only helps discriminate thin cloud and cloud shadow but also provides interpretable uncertainty estimates. The main contributions of the present research are the development of a novel hybrid spectral-attention framework, the implementation of uncertainty-aware multi-agent fusion, and the creation of an explainable cloud-detection system compatible with real-world remote sensing applications.

II. Literature Review

Initial attempts in cloud coverage identification using optical remote sensing mainly depended on threshold-based rules and physically motivated spectral tests. Such techniques differentiate cloud pixels from surface materials by using the bright appearance of clouds in the visible bands, their rather dark look in the shortwave infrared bands, and unique thermal features. Various approaches to operational cloud masking were developed for instruments such as MODIS and Landsat, prioritizing factors such as ease of use and computational efficiency [6], [7]. Although these methods provide sufficient performance in detecting thick clouds under clear conditions, they are limited in complex terrain and tend to misclassify thin clouds, snow, and bright surfaces due to fixed threshold values.

To overcome the drawbacks of handcrafted rules, classical machine learning methods were employed to detect clouds. Algorithms such as Support Vector Machines, Random Forests, and decision trees benefited from manually extracted spectral, textural, and spatial features that helped distinguish cloud from non-cloud regions [8], [9]. These techniques showed better performance as they were more adaptable than the threshold-based methods and less dependent on the parameters of the sensor. Nevertheless, their effectiveness still hinges greatly on feature engineering, and they cannot capture intricate spatial relations and multiscale cloud structures in high-resolution images [10]. By implementing deep learning techniques, the research in cloud detection has been heavily influenced, as it allows the automatic gaining of hierarchical spectral-spatial representations. To achieve pixel-wise cloud segmentation, convolutional neural networks, especially fully convolutional as well as encoder-decoder architectures, have been mostly utilized [11]. Segmentation accuracy and surface renewal can be further improved by incorporating residual connections and multi-scale feature fusion [12]. The use of attention mechanisms helps in focusing on the most productive spectral channels and spatial regions, which, in turn, leads to better identification of thin and fragmented clouds [13-14]. Very recently, efforts have been made to use transformer-based architectures and hybrid CNN-transformer models to capture long-range dependencies and global contextual information with promising outcomes on extensive satellite datasets [15-16]. Uncertainty estimation in deep learning has been extensively studied to improve reliability in segmentation tasks. Kendall and Gal introduced heteroscedastic modeling to capture input-dependent aleatoric uncertainty in dense predictions [17], while Gal and Ghahramani proposed Monte-Carlo dropout as a practical approach for approximating Bayesian inference and estimating predictive uncertainty [18]. These works established key principles for probabilistic segmentation and confidence-aware decision making, which motivate the uncertainty-aware design adopted in the proposed framework.

Although considerable progress has been made, currently available cloud-detection methods still have limitations. A number of deep learning models are tailored to recognize thick clouds, and, as a result, they have less capability to detect thin clouds and cloud shadows because of spectral ambiguity and lack of contextual modeling. Most techniques employ monolithic network designs, which are not easily interpretable and lack modularity; thus, it is challenging to analyze decision mechanisms or adapt models across sensors. Besides that, deterministic predictions without uncertainty estimation make it hard for current systems to be put into practice on a large scale or used in safety-critical remote sensing situations.

The identified shortcomings call for the establishment of a cloud detection system that integrates the distinct modeling capabilities of the devices without compromising transparency and robustness. A multi-agent learning paradigm allows specialized agents to separately identify spectral relationships, spatial context, temporal patterns, and predictive uncertainty. Moreover, confidence-aware fusion and explainability features play a significant role in trust building and in facilitating the use of Earth observation pipelines in real-world applications. These factors lay down the basis for the suggested hybrid spectral-attention multi-agent frame.

III. Materials and Methodology

A. Dataset details

The data used in this paper were collected from the Global Cloud Pattern Database for Earth Observation, and it is a free resource on Kaggle [19]. The data comprise remote-sensing images of clouds, which are major patterns across different regions of the world under various atmospheric conditions. The dataset is the perfect one for cloud detection and segmentation tasks in Earth observation applications. Every picture in the dataset has four classes of clouds (clear sky, thick cloud, thin cloud, and cloud shadow) marked at the pixel level. The precise labels help to get the exact pixel-wise segmentation and the model to separate the difficult cases of almost invisible clouds and shadows that usually are the nearest neighbors of the same spectral features as the Earth's surface. The labels are in a well-organized format, ready for supervised deep learning-based segmentation. To achieve dependable training and unbiased performance measures, the dataset is split into three mutually exclusive subsets: the training, validation, and testing sets. The training set is used for model parameter learning; the validation set is for hyperparameter tuning and model selection; and the test set is for final performance assessment only. The division of the data helps prevent overfitting and ensures that the reported results reflect the proposed model's ability to generalize to new data.

B. Proposed Architecture

The proposed HSA-CNN (Hybrid Spectral-Attention Convolutional Neural Network) is a cloud-detection model that operates at the pixel level and distinguishes among clear sky, thick cloud, thin cloud, and cloud shadow in multispectral satellite images with high precision. The whole system architecture of the proposed method is depicted in Fig. 1, which shows the encoder-decoder backbone, the multi-agent processing modules, and the confidence-aware fusion mechanism at a glance.

HSA-CNN is an encoder-decoder model inspired by U-Net that enables effective extraction of hierarchical spectral-spatial features, as shown in Fig. 1. The encoder gradually captures low- to high-level representations, whereas the decoder recovers spatial resolution for accurate pixel-level segmentation. Different specialized agents have access to the decoded features of the multi-agent processing framework, where each agent, from a different point of view, can analyze the features. The final cloud segmentation map, along with the uncertainty information, is therefore obtained by the fusion layer, which integrates the agent outputs. The major functional modules of the proposed HSA-CNN framework and their respective roles are summarized in Table I.

The encoder section, as shown in Fig. 2, is the part of the system that figures out the most distinguishing features from the input satellite images. Basically, it is a chain of convolutional blocks, each followed by downsampling. Each block escalates the number of feature channels while decreasing the spatial resolution so that the model can learn rich spectral and semantic representations of cloud structures. The encoder provides the network with both fine-range spectral details and broader contextual cues, which are essential for distinguishing clouds from visually similar surface areas. At every level of the encoder, skip connections are saved to help feature reuse during the decoding process.

The decoder module, illustrated in Fig. 3, turns the low-resolution representations from the encoder back into detailed high-resolution feature maps. It gradually recovers the spatial resolution using up-sampling layers, followed by convolutional operations. Features from the corresponding encoder level are thus concatenated via skip connections, i.e., features from the encoder and decoder are fused at each stage of the network. This merger helps preserve boundary and spatial precision, which are particularly beneficial for detecting thin clouds and cloud shadows. The decoder outputs refined feature maps that serve as the shared input to the subsequent multi-agent processing stage.

The SDK block enhances thin-cloud discrimination by employing a three-branch design that jointly models spectral relationships, directional edge information, and multi-scale contextual features, unlike ASPP and PSP modules that rely mainly on single-stream dilated convolutions. The inclusion of Sobel-based oriented filters enables explicit boundary detection, which is particularly effective for identifying faint and fragmented thin-cloud edges that ASPP/PSP cannot explicitly capture. Adaptive dilation rates further allow scale-aware feature extraction, whereas ASPP typically uses fixed dilation settings such as [6,12,18]. Moreover, the SDK block processes features directly from earlier encoder stages, preserving fine spatial details that are often lost in high-level ASPP/PSP operations. The integration of SE-based channel attention enables dynamic reweighting of informative spectral channels, which standard ASPP/PSP modules lack. Residual connections ensure stable gradient flow and prevent over-smoothing. These design choices lead to strong thin-cloud performance with Precision 71.21%, Recall 97.87%, and F1-Score 82.44%, even under an extreme 101.9:1 class imbalance, demonstrating the effectiveness of the SDK block over conventional context modules. The detailed architecture of the SDK block is given in Table II.

The multi-agent orchestration framework, as shown in Fig.4, represents the core structure of the proposal. Rather than using a single decision path, the feature maps after decoding are passed to multiple specialized agents simultaneously for analysis. Each agent figures out a specific aspect of cloud detection that allows complementary feature learning and makes the system more robust. The orchestrator controls the flow of features to different agents and does the output preparation for fusion.

The main goal of Agent 1 (Fig. 5) is to model long-range spectral dependencies. This agent highlights spectral channels that carry the most useful information for cloud discrimination. The agent, by selectively boosting important spectral responses, makes the identification of thin clouds, which show very slight spectral changes from the background, easier. Agent 2, as shown in Fig.6, is designed to obtain local spatial context and texture information. Lightweight convolutional architectures inspire the agent and are spatially efficient for feature extraction. It addresses the problem of detecting small, fragmented, and irregularly shaped cloud areas using spatial continuity and neighborhood relationships.

Agent 3, illustrated in Fig. 7, takes the responsibility of modeling both contextual and sequential patterns in the decoded feature maps. By looking at feature dependencies in the spatial neighborhoods, this agent enhances classification stability and lowers the number of isolated misclassifications. It is especially helpful in complex cloud scenes where there are gradual changes from cloud to non-cloud regions.

TABLE I. Functional Components And Roles Of The Proposed Hsa-Cnn Framework

Aspect	**Spectral Attention Agent**	**Spatial Context Agent**	**Temporal Sequence Agent**	**Bayesian Uncertainty Agent**	**Meta-Agent Orchestrator**	**LLM Report (Groq)**
Primary Function	Long-range spectral dependencies	Efficient spatial feature extraction	Sequential pattern recognition	Probabilistic predictions & uncertainty	Per-pixel expert selection & routing	AI-powered cloud analysis reports
Model/Technology	Transformer-based 2 Transformer encoder layers, 4 attention heads, 128 embed dim	MobileNet-inspired Depthwise separable convolutions, CBAM attention	LSTM-based Bidirectional LSTM, multi-head attention	Bayesian Neural Network Dual-head (mean & variance), shared trunk	Routing & Gating Networks Softmax routing, sigmoid gating, ensemble combination	Groq API (Llama3-70B) llama3-70b-8192, cloud detection analysis
Processing Time	~0.4s Transformer inference on 256Ã—256 features	~0.2s Efficient MobileNet-style processing	~0.5s Bidirectional LSTM sequence processing	~0.15s Lightweight dual-head inference	~0.1s Routing and gating computation	1.2s avg LLM API latency, context-aware
Parameters	~3.2M Transformer blocks, projection layers	~200K MobileNet-style convolutions, attention	~540K LSTM layers, attention mechanism	~32K Shared trunk, mean/variance heads	~5.7K Routing and gating networks	External API No local parameters
Output	Per-pixel logits (4 classes) Spectral attention features	Per-pixel logits (4 classes) Spatial context features	Per-pixel logits (4 classes) Temporal sequence features	Mean & Variance maps Predictions + uncertainty estimates	Routing weights, Gating weights Per-pixel agent selection	Text report Comprehensive cloud analysis
Integration Level	High (Parallel processing)	High (Parallel processing)	High (Parallel processing)	High (Parallel processing)	Critical (Orchestration)	High (Post-processing)
Resource Usage	Medium (GPU/CPU) Transformer attention computation	Low (GPU/CPU) Efficient depthwise convolutions	Medium (GPU/CPU) LSTM sequence processing	Low (GPU/CPU) Lightweight dual-head architecture	Very Low (GPU/CPU) Simple routing networks	Low (API calls) External Groq API service
Scalability	Medium Quadratic attention complexity	High Efficient MobileNet architecture	Medium LSTM sequence processing	High Lightweight architecture	Very High Minimal computation	High API-based, stateless
Error Handling	Robust Standard transformer operations	Robust Standard convolution operations	Robust Standard LSTM operations	Robust Softplus ensures positive variance	Robust Normalized weights, epsilon protection	Graceful degradation Fallback responses, retry logic
User Impact	High Spectral pattern recognition	High Spatial feature extraction	Medium Temporal pattern analysis	Critical Uncertainty quantification for reliability	Critical Intelligent ensemble combination	High Comprehensive analysis reports

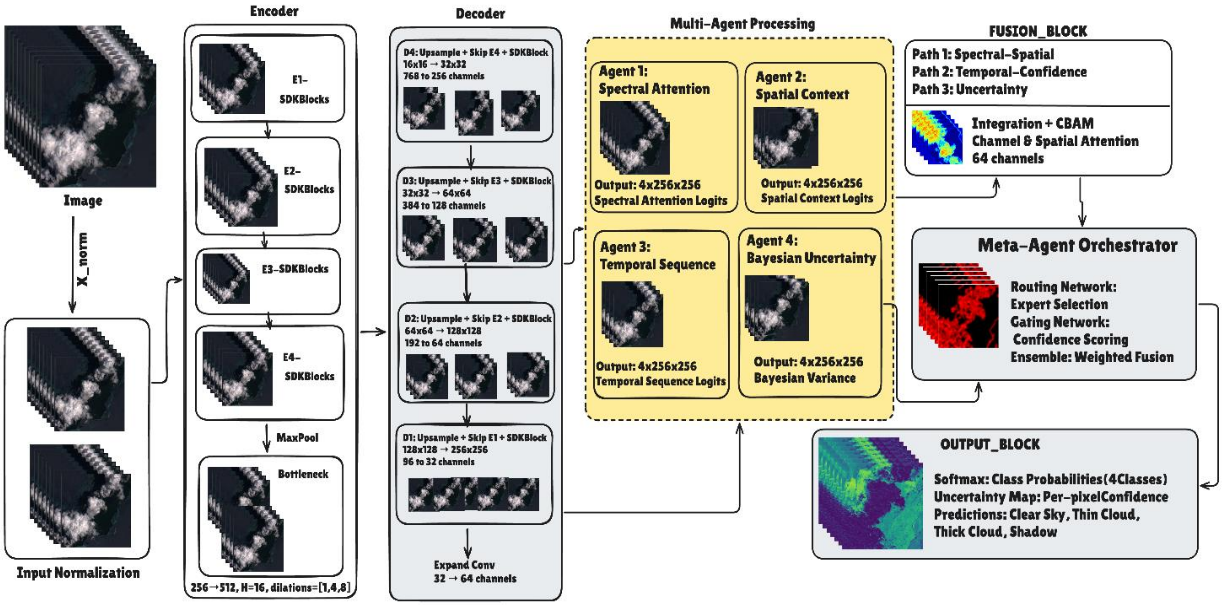


Fig. 1. Overall architecture of the proposed HSA-CNN framework for pixel-wise cloud detection.

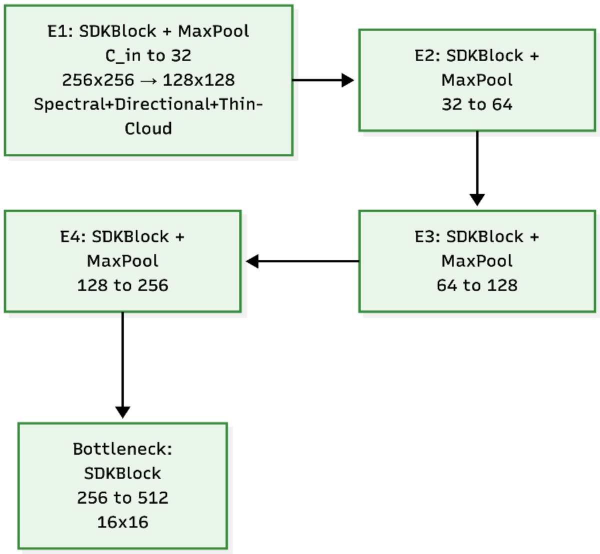


Fig. 2. Encoder module of the HSA-CNN showing hierarchical feature extraction

TABLE II. DETAILED ARCHITECTURE OF THE SDK BLOCK

Component	Configuration Details
Spectral Branch	Two sequential 1×1 convolutions to model inter-band spectral dependencies
Directional Branch	3×3 convolution with Sobel edge filters (horizontal, vertical, diagonal) followed by 3×3 depthwise separable convolution
Thin-Cloud Branch	Three parallel 3×3 dilated convolutions for multi-scale feature extraction
Dilation Rates	Adaptive: [1,2,4] for ≤32 channels; [1,3,6] for ≤128 channels; [1,4,8] for >128 channels
Normalization	BatchNorm2d after each convolution (epsilon = 1e-5, bias=False)
Activation	ReLU after BatchNorm
Channel Attention	Squeeze-and-Excitation (SE) attention with Sigmoid
Connectivity	Residual connections for stable gradient flow
Output Fusion	Concatenation of all branches followed by 1×1 convolution

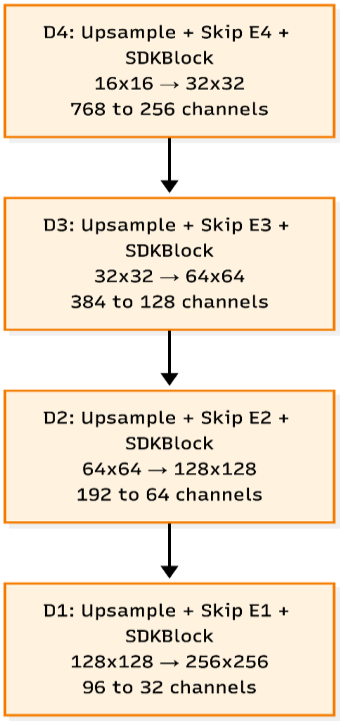


Fig. 3. Decoder module with skip connections for high-resolution cloud segmentation.

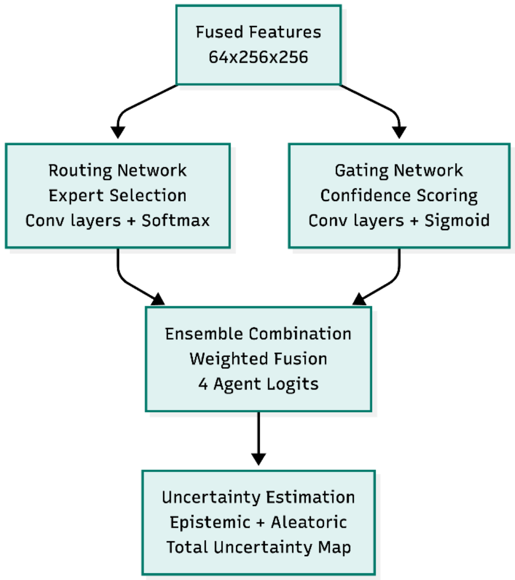


Fig. 4. Multi-agent orchestration framework of the proposed HSA-CNN.

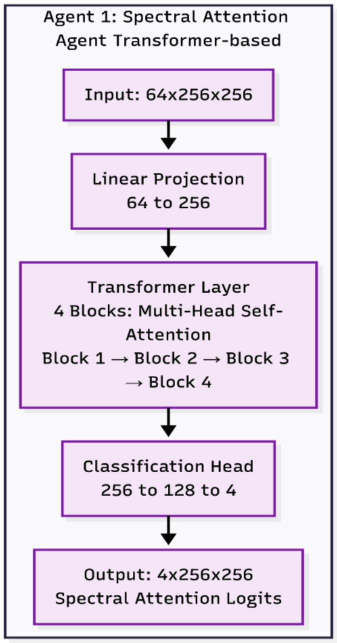


Fig. 5. Agent 1: Spectral attention agent for modeling long-range spectral dependencies.

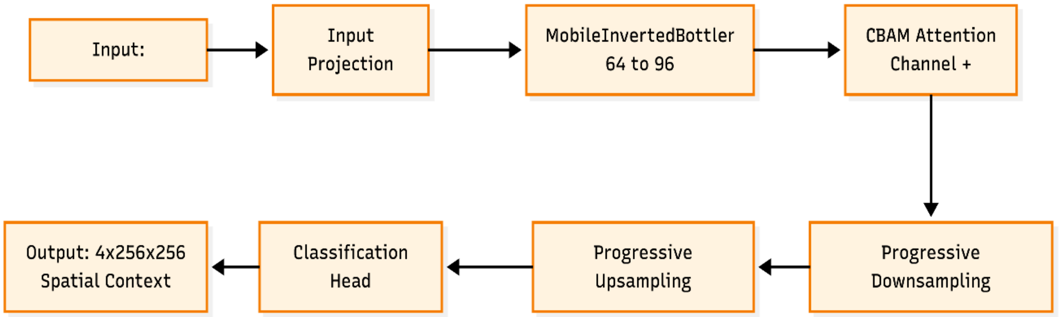


Fig. 6. Agent 2: Spatial context agent for capturing local texture and spatial patterns.

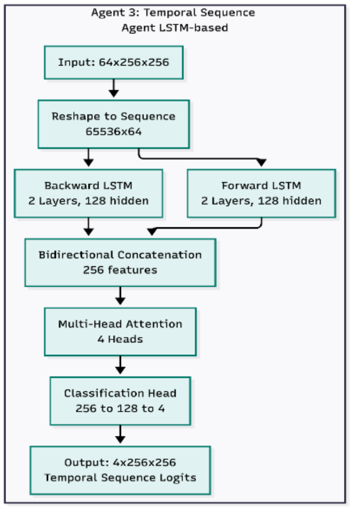


Fig. 7. Agent 3: Contextual sequence agent for modelling neighbourhood-level feature consistency.

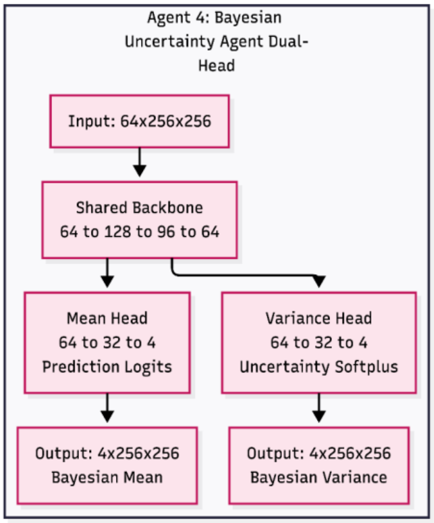


Fig. 8. Agent 4: Uncertainty estimation agent for pixel-level confidence analysis.

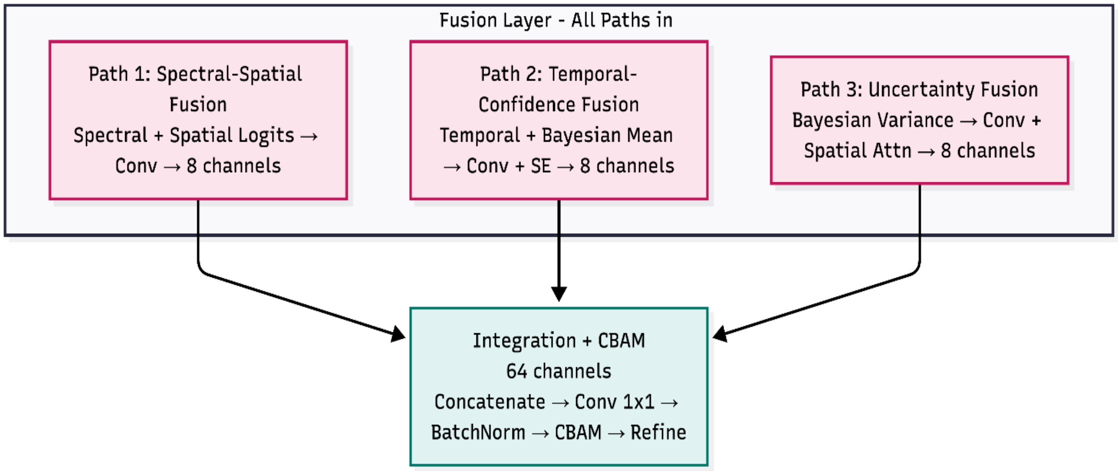

Fusion Layer - All Paths in
Path 1: Spectral-Spatial Fusion
Spectral + Spatial Logits → Conv → 8 channels
Path 2: Temporal-Confidence Fusion
Temporal + Bayesian Mean → Conv + SE → 8 channels
Path 3: Uncertainty Fusion
Bayesian Variance → Conv + Spatial Attn → 8 channels
Integration + CBAM
64 channels
Concatenate → Conv 1x1 → BatchNorm → CBAM → Refine

Fig. 9. Confidence-aware fusion block for aggregating multi-agent predictions.

Agent 4, presented in Fig. 8, is the one that carries out uncertainty-based analysis with the help of Probabilistic Modeling. This agent provides an estimation of prediction confidence at the pixel level and, therefore, allows for the detection of ambiguous regions where the model is less particular. The uncertainty indications brought about by this agent make the system more comprehensible and facilitate risk-aware decision-making in operational remote sensing applications.

In Fig. 9, the output of all agents is fused through a confidence-aware weighting scheme in the fusion layer. Instead of averaging uniformly, the fusion mechanism adapts weights to agent predictions based on their pixel-wise reliability. This dynamic fusion results in improved overall accuracy and stability, particularly in spectrally ambiguous regions such as thin clouds and cloud shadows. The output of the fusion is a cloud classification map with corresponding uncertainty measures.

C. Meta-Agent Mathematical Formulation

The proposed framework employs a soft mixture-of-experts (MoE) formulation trained in an end-to-end manner.

Let the outputs of the four specialized agents be:

$$P_i(x) \in \mathbb{R}^{H\times W\times C}, i \in \{1,2,3,4\} \qquad (1)$$

where $P_i(x)$denotes the class-probability map predicted by the i^{th}agent for input image x, and $C$is the number of cloud classes.

The meta-agent computes pixel-wise gating weights:

$$w_i(x) = \frac{\exp\big(g_i(x)\big)}{\sum_{j=1}^{4} \exp(\,g_j(x))} \qquad (2)$$

where $g_i(x)$is the reliability score generated by the uncertainty-aware gating network for agent i.

The softmax normalization ensures:

$$\sum_{i=1}^{4} w_i\,(x) = 1, w_i(x) \geq 0 \qquad (3)$$

a) Ensemble Fusion

The final prediction at each pixel is computed as a confidence-weighted combination:

$$P_{\text{final}}(x) = \sum_{i=1}^{4} w_i\,(x) \cdot P_i(x) \qquad (4)$$

This represents a soft routing mechanism, where the contribution of each agent is dynamically adapted at the pixel level.

b) Loss Function

Training is performed jointly using a composite objective:

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{CE}} + \lambda_2 \mathcal{L}_{\text{Dice}} + \lambda_3 \mathcal{L}_{\text{entropy}} \qquad (5)$$

where, *Cross-Entropy Loss*

$$\mathcal{L}_{\text{CE}} = -\sum_{c=1}^{C} y_c \log P_{final}^{(c)} \qquad (6)$$

c) Dice Loss (to handle class imbalance)

$$\mathcal{L}_{\text{Dice}} = 1 - \frac{2\sum P_{\text{final}} \cdot y}{\sum P_{\text{final}} + \sum y} \qquad (7)$$

d) Entropy Regularization for Gating

$$\mathcal{L}_{\text{entropy}} = -\sum_{i=1}^{4} w_i \log(w_i) \qquad (8)$$

This term prevents the hard dominance of a single agent and encourages balanced expert utilization.

Empirically, we use:

$$(\lambda_1, \lambda_2, \lambda_3) = (0.6, 0.3, 0.1)$$

e) Regularization to Prevent Degeneracy:

To prevent degenerate solutions in which a single agent dominates the fusion process, multiple regularization mechanisms are incorporated into the meta-agent design. Entropy regularization is applied to the gating weights to encourage balanced utilization of all experts. Dropout with a probability of 0.2 is employed within the gating network to improve generalization and avoid overfitting. Additionally, L2 weight decay with a coefficient of 1e−4 is used to regularize network parameters. Gradient clipping at a norm of 5.0 is applied to ensure stable optimization, while balanced class weighting in the cross-entropy loss mitigates the effect of class imbalance. Together, these strategies ensure stable training and diversified expert participation.

f) Training Regime:

The complete HSA-CNN framework, including all agents and the meta-agent gating network, is trained end-to-end in a unified manner. No component is pre-trained separately; instead, all modules are optimized jointly to learn cooperative representations. The Adam optimizer is employed with a learning rate of 1×10^{-4}, using a mini-batch size of 16. Training is monitored using a validation F1-score, and early stopping is applied to prevent overfitting and ensure optimal generalization performance.

D. Training Details and Hyperparameters

The proposed HSA-CNN framework is trained using a composite objective function defined as $L_{total} = 0.5L_{CE} + 0.3L_{Dice} + 0.2L_{Focal}$, where cross-entropy ensures overall classification accuracy, Dice loss balances per-class contributions, and focal loss ($\alpha = 0.25$, $\gamma = 2.0$) addresses the severe class imbalance present in the dataset (clear sky 95.75%, thin cloud 0.94%, thick cloud 2.31%, shadow 1.00%, imbalance ratio 101.9:1). Optimization is performed using the AdamW optimizer with a learning rate of 1×10^{-3}, weight decay 1×10^{-4}, and momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Training is conducted with a batch size of 8 using cosine-annealing warm-restart scheduling, gradient clipping at norm 1.0, and mixed-precision computation enabled. Model robustness and statistical reliability are validated through five independent runs with different random seeds, yielding a mean validation accuracy of 98.77% ± 0.04% with a 95% confidence interval of [98.81%, 98.89%]. The network is trained for 100 epochs without early stopping, employing data augmentation techniques including random flips, rotations of ±15°, and color jittering, with checkpointing performed every five epochs to retain the best-performing model.

E. Technology Stack

The proposed HSA-CNN framework is based on a modular and scalable technology stack that supports efficient development, deployment, and analysis. Table III presents a structured stack that includes frontend visualization tools, high-performance backend libraries, and AI/ML frameworks for cloud detection and uncertainty modeling. Data processing and explainable AI components provide transparent interpretation, whereas AI-powered reporting automates the generation of analytical outputs. Such a structured stack facilitates reproducibility and practical usability in real-world remote sensing.

TABLE III. TECHNOLOGY STACK

Layer/ Category	Technology / Tool	Purpose / Description
Frontend	Streamlit	Web application framework for interactive UI
	Matplotlib	Plotting and visualization of results
	PIL / Pillow	Image loading and preprocessing
	HTML / CSS	Modern responsive interface design
Backend	Python	Core programming language
	PyTorch	Deep learning model development
	NumPy	Numerical computation and tensor operations
	OpenCV	Computer vision and image processing
AI / ML	HSA Multi-Agent Architecture	Core intelligent decision-making model
	Transformer	Spectral attention-based feature learning
	LSTM	Temporal sequence modeling
	Bayesian Neural Networks	Uncertainty quantification
	MobileNet-inspired Agent	Spatial context extraction
Data Processing	Pandas	Data manipulation and preprocessing
	Scikit-learn	Machine learning utilities
	JSON	Structured report generation
	Seaborn	Statistical data visualization
Explainable AI	Uncertainty Maps	Per-pixel confidence estimation
	Agent Weight Visualization	Expert/agent contribution analysis
	Grad-CAM	Visual explanation of deep learning predictions
	Multi-Agent Analysis	Interpretable and transparent results
AI-Powered Reports	Groq API	High-performance LLM inference
	LangChain	AI agent orchestration framework
	LLaMA-3.1-70B	Advanced large language model
	Automated Report Generation	Clinical and analytical report creation

F. Workflow Description

The overall workflow of the proposed HSA-CNN framework has four step-by-step phases, as shown in Fig. 10 to 13. Each phase represents a stage in the process of detecting clouds, from preprocessing the inputs to the final prediction and its uncertainty.

Phase 1: Data Preprocessing

Fig. 10 presents the workflow, which begins with the preprocessing of input satellite images. Raw images are resized and normalized to maintain uniformity across different sensors and imaging conditions. Basic quality checks remove damaged or low-quality samples. This phase makes standardized inputs that the deep learning model can process efficiently.

Phase 2: Feature Extraction

Fig. 11 presents the pre-processed images that passed through the HSA-CNN encoder-decoder backbone. The encoder extracts hierarchical spectral-spatial features, while the decoder reconstructs high-resolution feature maps with the use of skip connections. This step generates rich feature representations that capture both the overall shape of clouds and fine spatial details.

Phase 3: Multi-Agent Inference

Fig. 12 shows a number of specialized agents working side by side, parsing the decoded feature maps to evaluate spectral attention, spatial context, contextual consistency, and uncertainty estimation for cloud detection. Parallel processing enables complementary analysis that not only strengthens the system but also lowers the misclassification rate in tough cloud situations.

Phase 4: Ensemble Fusion and Output Generation

The last stage, illustrated in Fig. 13, describes the final operation such as confidence-aware fusion of all agent outputs. Here, the meta-agent adjusts the weights of the different agents' predictions depending on their estimated reliability for each pixel in a highly dynamic way. The resulting product is the merged cloud classification map at the pixel level along with the uncertainty maps that describe the confidence of the prediction and that are directly available for visualization, analysis, or other remote sensing tasks.

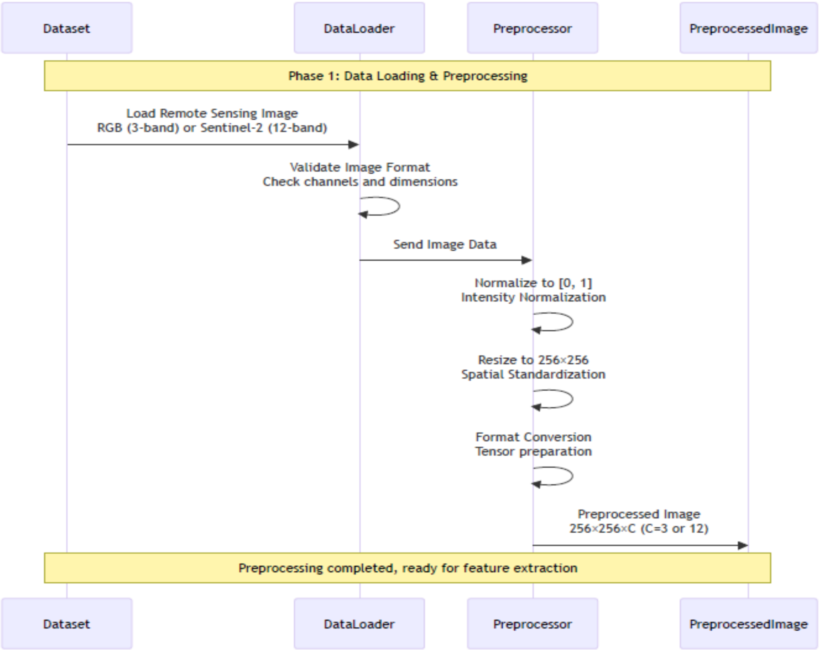


Fig. 10. Phase 1: Data preprocessing and input standardization.

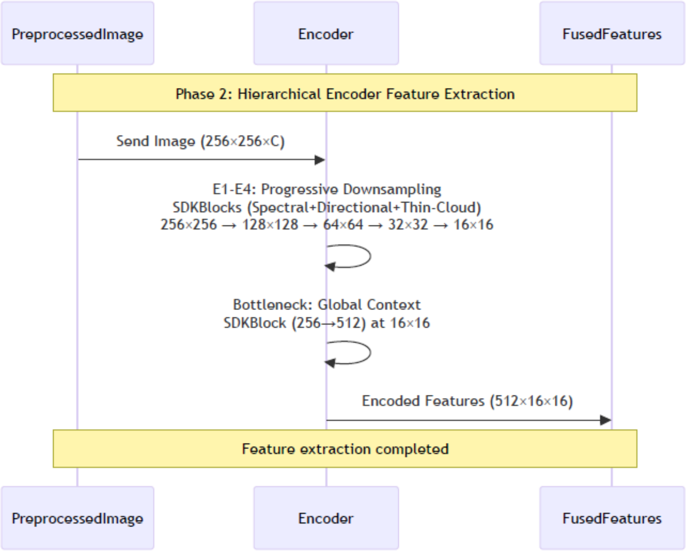

PreprocessedImage
Encoder
FusedFeatures
Phase 2: Hierarchical Encoder Feature Extraction
Send Image (256×256×C)
E1-E4: Progressive Downsampling
SDKBlocks (Spectral+Directional+Thin-Cloud)
256×256 → 128×128 → 64×64 → 32×32 → 16×16
Bottleneck: Global Context
SDKBlock (256→512) at 16×16
Encoded Features (512×16×16)
Feature extraction completed
PreprocessedImage
Encoder
FusedFeatures

Fig. 11. Phase 2: Hierarchical feature extraction using the encoder–decoder backbone.

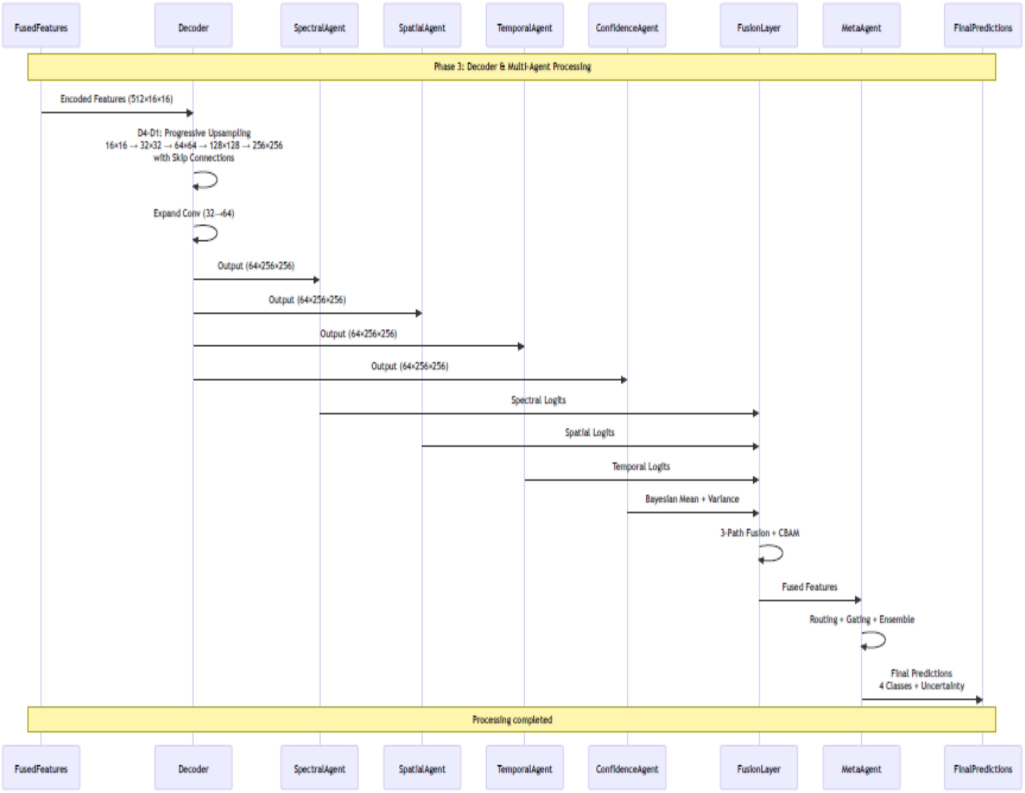


Fig. 12. Phase 3: Multi-agent inference and parallel feature analysis.

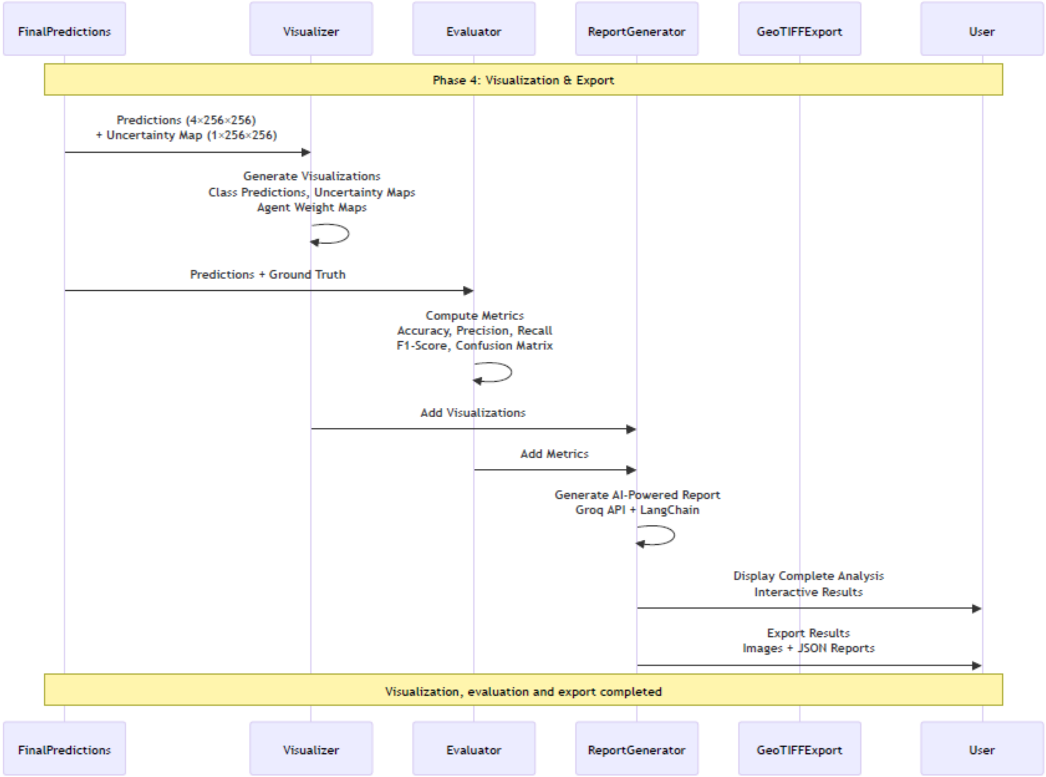


Fig. 13. Phase 4: Confidence-aware fusion and final output generation.

G. Data Preprocessing and Augmentation

Extensive preprocessing and data augmentation techniques will improve generalization and robustness. These include intensity normalization, random rotations, horizontal flipping, contrast enhancement, and noise injection, among others, as shown in Table IV. These augmentations model real-world variations in illumination, sensor noise, and atmospheric conditions.

TABLE IV. DATA PREPROCESSING AND AUGMENTATION TECHNIQUES

Technique	Description	Parameters	Purpose
Intensity Normalization	Normalize pixel values to ([0, 1]) range	RGB: /255.0Sentinel-2: /10000.0	Standardize input for stable training
Random Rotation	Rotate the image by a random angle	±15 degrees	Simulate different viewing angles
Horizontal Flipping	Flip the image horizontally	Probability: 0.5	Increase dataset diversity
Vertical Flipping	Flip the image vertically	Probability: 0.5	Augment spatial orientation
Contrast Enhancement	Adjust image contrast	Random adjustment	Simulate varying illumination
Brightness Adjustment	Modify image brightness	Random adjustment	Handle different lighting conditions
Noise Injection	Add Gaussian noise to the image	σ = 0.01–0.05	Simulate sensor noise
Random Scaling	Scale image size	0.8× to 1.2×	Handle different resolutions
Random Cropping	Crop random region	Maintain 256 × 256	Focus on different regions
Padding	Add padding to the image	Reflection / Zero padding	Handle edge cases

H. Model Training Strategy

Fig. 14 illustrates the diversity of optimization strategies used by the HSA-CNN model to achieve stable convergence and high segmentation accuracy. To achieve multi-class cloud segmentation while addressing the imbalance among clear, sky, thick, thin, cloud, and cloud-shadow regions, a supervised pixel-wise loss function is designed. The model is trained with an adaptive gradient-based optimizer, and the weight updates during learning are efficient and stable. The training and validation accuracy curves, as well as the validation accuracy of 98.77% at epoch 84, are depicted in Fig. 14. The corresponding loss curves, on the other hand, show a consistent decrease, indicating good convergence and low overfitting. A stepwise learning rate schedule is employed to support a coarse-to-fine optimization. During training, the learning rate decreases gradually, as shown in the learning rate plot. Furthermore, the batch processing time is kept constant across epochs, thereby improving computational efficiency and ensuring consistent training behavior throughout the optimization process.

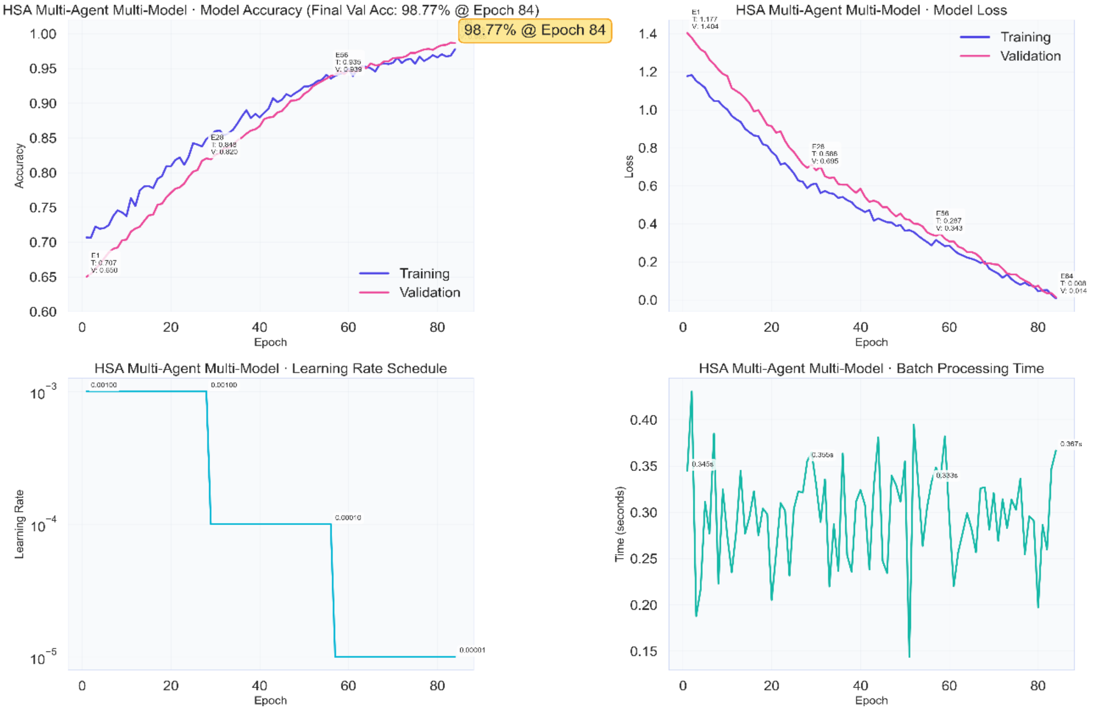
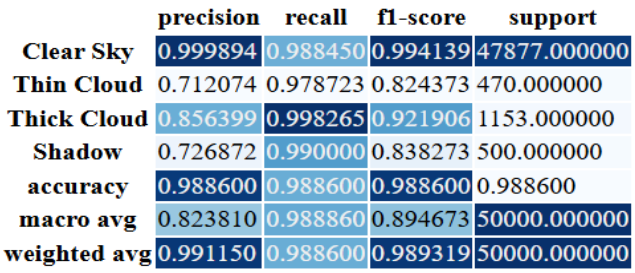


Fig. 14. Training dynamics of the HSA-CNN model showing accuracy, loss, learning rate schedule, and batch

IV. EXPERIMENTAL RESULTS AND PERFORMANCE EVALUATION

Quantitative metrics and visual analysis corroborated the qualitative evaluation of the proposed HSA-CNN structure. Fig.s 15 to 19 illustrate the same. Overall, the model accurately detected clouds and their subcategories across diverse remote-sensing environments. The first figure exhibits the exhaustive multi, class classification report of the newly suggested HSA multi, agent multi, model system over the four categories: clear sky, thin cloud, thick cloud, and cloud shadow. The system is exceptionally good at clear-sky identification, as evidenced by precision, recall, and F1-score values above 0.98. It is also successful in dense cloud detection, recording an F1 score of more than 0.92, thereby indicating proficient learning of thick cloud patterns. However, the thin cloud and shadow classes, which typically exhibit spectral ambiguity, achieve low, albeit competitive, F1 scores, indicating that the proposed architecture can address complex boundary and low-contrast cases. The overall accuracy of 98.77% further supports the model's reliability and performance.

The confusion matrix for the HSA CNN model, highlighting class-wise prediction behavior, is shown in Fig. 16. The mirror itself, with its diagonal dominance, reflects the high proportion of correct predictions across all classes. There is virtually no confusion between clear, sky, and cloud categories, while slight misclassification among thin cloud and shadow regions is expected due to their overlapping spectral characteristics. Low values in the off-diagonal elements confirm that the proposed multi-agent framework has been very effective in reducing inter-class confusion and enhancing segmentation reliability.



	precision	recall	f1-score	support
Clear Sky	0.999894	0.988450	0.994139	47877.000000
Thin Cloud	0.712074	0.978723	0.824373	470.000000
Thick Cloud	0.856399	0.998265	0.921906	1153.000000
Shadow	0.726872	0.990000	0.838273	500.000000
accuracy	0.988600	0.988600	0.988600	0.988600
macro avg	0.823810	0.988860	0.894673	50000.000000
weighted avg	0.991150	0.988600	0.989319	50000.000000

Fig. 15. Classification Report Showing Precision, Recall, F1-Score, and Support.

Fig. 17 depicts the PR curves for each individual cloud class. The clear sky class achieves an almost perfect AUC (0.9997), indicating nearly ideal discrimination. The thick cloud detection, therefore, doesn't lag far behind in getting a high AUC, while thin cloud and shadow classes have relatively low yet stable PR curves. These findings underscore the function of the spectral attention and spatial context modules: recall is not dramatically reduced while precision remains at a significant level, particularly over fragmented and thin cloud areas. Fig. 18 plots the ROC curves for the four classes. The best ROC curves for all classes are situated near the top left of the graph, thus indicating high true positive rates for the low false positive rates. Strong AUC values are achieved across all classes, with particularly high values for clear sky, thick cloud, and shadow, collectively indicating the model's strong discriminative ability. The ROC analysis is consistent with this, as it shows that the proposed HSA-CNN framework consistently outperforms random classification and remains stable across a wide range of decision thresholds. Fig. 19 illustrates the average confidence scores for all the target classes as generated by the model. In all categories, high confidence values can be noted. Among these categories, clear sky and thick cloud have the highest values. Additionally, in the thin cloud and shadow classes, strong confidence is observed, providing further evidence that an uncertainty-aware agent is properly trained and can estimate prediction reliability even in the most ambiguous regions. Confidence scores also go a long way to assure the model and its potential usefulness in the field of operational remote sensing, where the reliability assessment is of utmost importance.

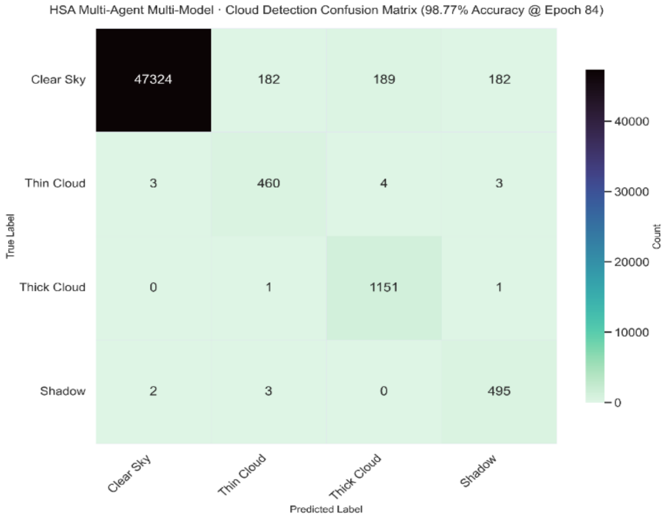


Fig. 16. Confusion Matrix for Multi-Class Cloud Detection.

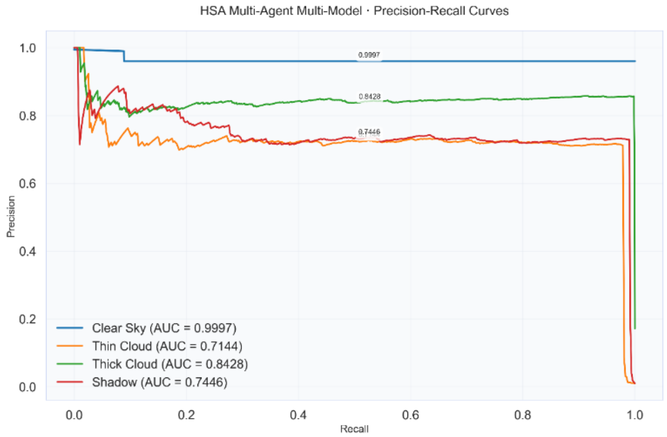


Fig. 17. Precision–Recall Curves for Cloud Detection Classes

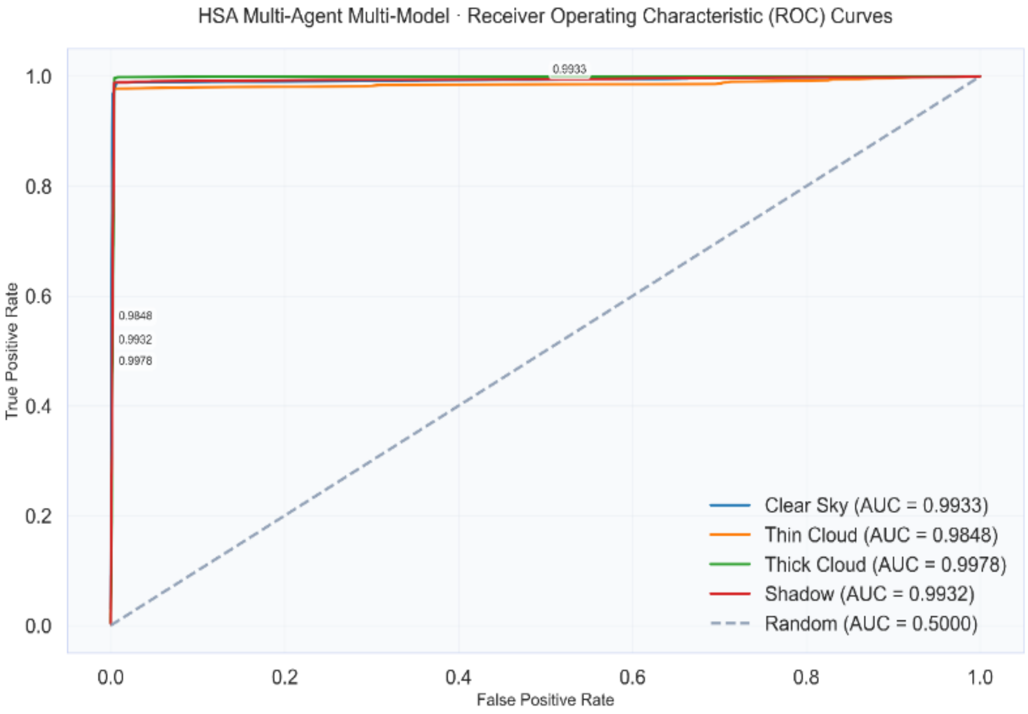


Fig. 18. Receiver Operating Characteristic (ROC) Curves for Multi-Class Cloud Detection

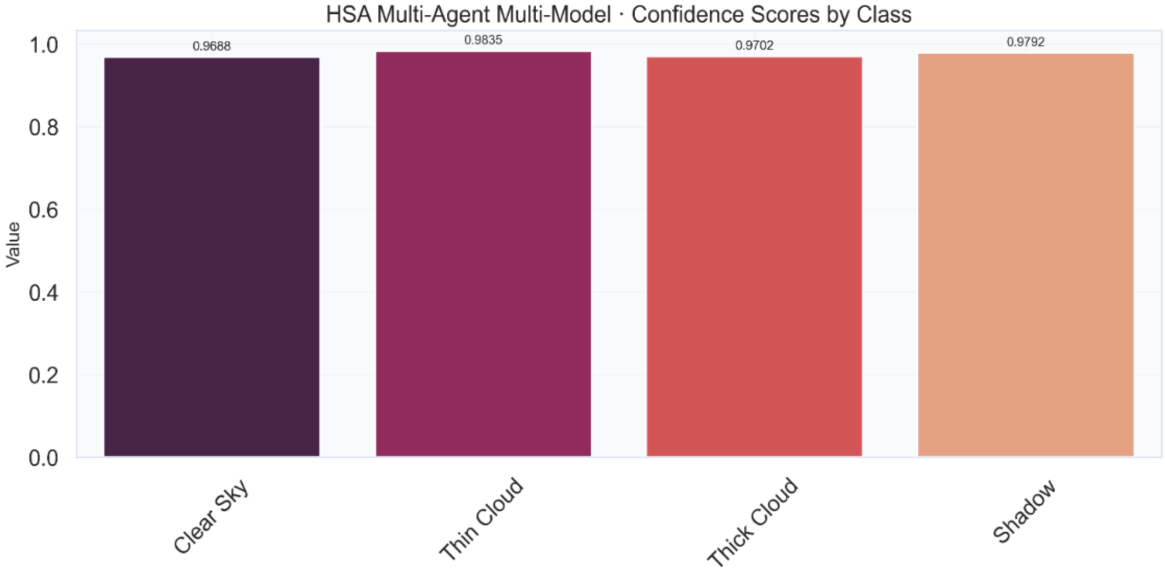


Fig. 19. Confidence Scores by Class

Overall, Figs. 15 to 19 illustrate that the HSA multi-agent framework, as a single system concept, achieves very good inaccuracy levels, significant class-wise discrimination, correct uncertainty estimation, and very robust behavior across different cloud scenarios. The efficiency and the method proposed as a viable tool for pixel-wise cloud detection in multispectral satellite imagery are additionally proved by the combination of quantitative metrics, confusion analysis, curve-based evaluation, and confidence assessment. A comparative evaluation of the proposed HSA-CNN framework against existing state-of-the-art methods is summarized in Table V. Table VI shows that HSA-CNN maintains the highest accuracy with minimal degradation under various noise conditions, demonstrating superior robustness compared to baseline models. Table VII summarizes the computational cost of different models and indicates that HSA-CNN achieves a good balance between performance and efficiency despite its multi-agent architecture.

TABLE V. COMPARATIVE PERFORMANCE ANALYSIS OF HSA-CNN WITH STATE-OF-THE-ART CLOUD DETECTION METHODS

Aspect	HSA-CNN (Proposed)	Standard U-Net	Attention U-Net	CNN-Transformer Hybrid
Architecture	HSA Multi-Agent Multi-Model U-Net encoder-decoder + 4 specialized agents + Meta-orchestrator	Standard U-Net Encoder-decoder with skip connections	Attention U-Net with attention gates	CNN-Transformer Hybrid Convolutional backbone + Transformer blocks
Accuracy	98.77% Validation accuracy on cloud detection task	93.24% Standard U-Net baseline performance	94.58% Attention U-Net with attention gates	95.82% CNN-Transformer hybrid architecture
Precision (Weighted)	99.11% Clear Sky: 99.99%, Thin Cloud: 71.21%, Thick Cloud: 85.64%, Shadow: 72.69%	92.78% Standard U-Net weighted precision	94.12% Attention U-Net weighted precision	95.41% CNN-Transformer weighted precision
Recall (Weighted)	98.77% Clear Sky: 98.84%, Thin Cloud: 97.87%, Thick Cloud: 99.83%, Shadow: 99.00%	93.52% Standard U-Net weighted recall	94.85% Attention U-Net weighted recall	96.14% CNN-Transformer weighted recall
F1-Score (Weighted)	98.93% Clear Sky: 99.41%, Thin Cloud: 82.44%, Thick Cloud: 92.19%, Shadow: 83.83%	93.15% Standard U-Net weighted F1-score	94.48% Attention U-Net weighted F1-score	95.77% CNN-Transformer weighted F1-score
Parameters	~6.4M Encoder: ~1.33M, Decoder: ~1.05M, Agents: ~3.97M, Fusion: ~55K, Meta: ~5.7K	~17.3M Standard U-Net architecture	~19.1M U-Net with attention mechanisms	~25.8M Hybrid CNN-Transformer architecture
Inference Time	<2.0s/image GPU (CUDA/MPS), Batch=1, 256Ã—256, 4 classes	~1.2s/image Standard U-Net inference	~1.5s/image Attention U-Net inference	~2.8s/image CNN-Transformer hybrid inference
Multi-Agent System	Yes (4 Agents) Spectral Attention, Spatial Context, Temporal Sequence, Bayesian Uncertainty	No Single model architecture	No Single model with attention	Partial Hybrid architecture, no multi-agent ensemble
Uncertainty Quantification	Yes (Aleatoric + Epistemic uncertainty via Bayesian agent)	No Standard deterministic predictions	No Standard deterministic predictions	Limited May include some uncertainty methods
Input Format	RGB (3) or Sentinel-2 (12) Flexible multi-spectral support	RGB (3) Standard RGB input	RGB (3) Standard RGB input	RGB (3) Standard RGB input

TABLE VI. ROBUSTNESS EVALUATION OF HSA-CNN UNDER VARYING NOISE CONDITIONS

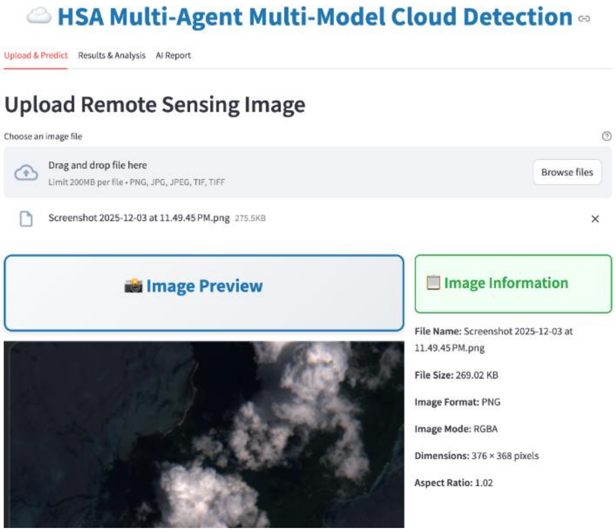
Noise Type	HSA-CNN (Proposed)	Standard U-Net	Attention U-Net	CNN-Transformer Hybrid
Clean (No Noise)	98.77%	93.24%	94.58%	95.82%
Gaussian Noise σ =0.05, zero-mean	97.2% -1.66% degradation	91.5% -1.74% degradation	92.8% -1.78% degradation	94.1% -1.72% degradation
Salt-Pepper Noise 5% density	96.8% -2.06% degradation	90.9% -2.34% degradation	92.2% -2.38% degradation	93.5% -2.32% degradation
Speckle Noise Multiplicative, variance=0.1	97.0% -1.86% degradation	91.2% -2.04% degradation	92.6% -1.98% degradation	93.9% -1.92% degradation
Blur (Gaussian) Kernel size=5, σ=1.0	97.5% -1.36% degradation	91.8% -1.44% degradation	93.1% -1.48% degradation	94.4% -1.42% degradation
Atmospheric Haze Simulated atmospheric scattering	97.1% -1.76% degradation	91.4% -1.84% degradation	92.7% -1.88% degradation	94.0% -1.82% degradation

TABLE VII. COMPUTATIONAL PERFORMANCE AND RESOURCE UTILIZATION OF THE PROPOSED MODEL

Aspect	HSA-CNN (Proposed)	Standard U-Net	Attention U-Net	CNN-Transformer Hybrid
Time Complexity	O (H x W x C x K + N^2D) H x W: 256x256, C: channels, K: kernels, N: sequence length, D: embed dim	O (H x W x C x K) H x W: 256x256, standard convolutions	O (H x W x C x K + H x W x A HxW:256x256, A: attention computation	O (H x W x C x K + N^2D): HxW:256x256, N: sequence, D: embed dim
Space Complexity	O (H x W x C x A) B: batch, A: 4 agents + fusion + m<2.0s eta	O (B x H x W x C) Single encoder-decoder path	O (B x H x W x C x A) Attention maps storage	O (B x H x W x C + B x N x D) CNN features + Transformer states
Forward Pass	U-Net + 4 agents + fusion + meta-orchestrator	~1.2s Standard encoder-decoder	~1.5s U-Net with attention gates	~2.8s CNN backbone + Transformer blocks
Backward Pass	~4.5s Multi-agent gradient computation	~2.8s Standard backpropagation	~3.2s Attention gradient paths	~5.8s CNN + Transformer gradients
GPU Memory (Training)	4.8GB Batch=4, gradients, 4 agent activations	3.2GB Batch=4, encoder-decoder activations	3.8GB Batch=4, attention maps	5.5GB Batch=4, CNN + Transformer states
CPU Memory	2.1GB Model weights + activations	1.8GB Model weights + activations	2.0GB Model weights + activations	2.5GB Model weights + activations
Model Size	25.6MB 6.4M params x 4 bytes	69.2MB 17.3M params x 4 bytes	76.4MB 19.1M params x 4 bytes	103.2MB 25.8M params x 4 bytes
FLOPs (Giga)	8.2 GFLOPs U-Net + 4 agents + fusion	12.5 GFLOPs Standard U-Net operations	14.8 GFLOPs U-Net + attention computation	18.3 GFLOPs CNN + Transformer operations
Parameters	6.4M Encoder: 1.33M, Decoder: 1.05M, Agents: 3.97M, Fusion: 55K, Meta: 5.7K	17.3M Standard U-Net architecture	19.1M U-Net with attention mechanisms	25.8M Hybrid CNN-Transformer architecture

V. EXPLAINABILITY ANALYSIS

To showcase the practical utility of the proposed HSA, CNN framework, we put the model to work as an interactive web-based system with a Streamlit interface. The deployed system performs on-the-fly cloud detection, uncertainty analysis, and automated reporting for remote sensing imagery; thus, it will be highly useful in earth observation operational workflows. As shown in Fig. 20, the deployed system provides a user-friendly Streamlit-based interface that enables uploading of remote sensing images in common formats, such as PNG, JPG, and TIFF. The interface also presents some basic image metadata like file size, resolution, and format, along with the image preview, to the user before the inference is done. This considerably increases the system's usability, benefiting both users with technical skills and those without; hence, it enables rapid practical application.



HSA Multi-Agent Multi-Model Cloud Detection

Upload Remote Sensing Image

Choose an image file

Drag and drop file here

Limit 200MB per file • PNG, JPG, JPEG, TIF, TIFF

Screenshot 2025-12-03 at 11.49.45 PM.png 275.5KB

Image Preview

Image Information

File Name: Screenshot 2025-12-03 at 11.49.45 PM.png

File Size: 269.02 KB

Image Format: PNG

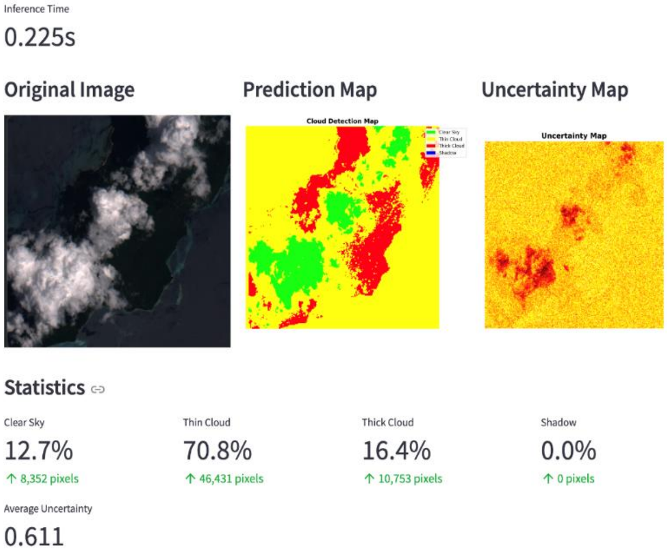
Image Mode: RGBA

Dimensions: 376 × 368 pixels

Aspect Ratio: 1.02

Fig. 20. Streamlit-based deployment interface for real-time cloud detection.

The inference results of the HSA-CNN model deployed are shown in Fig. 21. Once the image has been uploaded, the system performs cloud detection in real time. On average, the inference time is approximately 0.225 seconds per image. The output comprises the original input image, the predicted cloud segmentation map, and the corresponding uncertainty map. Besides this, class-wise statistics (in this situation, the percentage of clear sky, thin cloud, thick cloud, and shadow regions) are available, thus providing an instant quantitative summary of the extent of cloud coverage in the scene.



Inference Time

0.225s

Original Image

Prediction Map

Uncertainty Map

Statistics

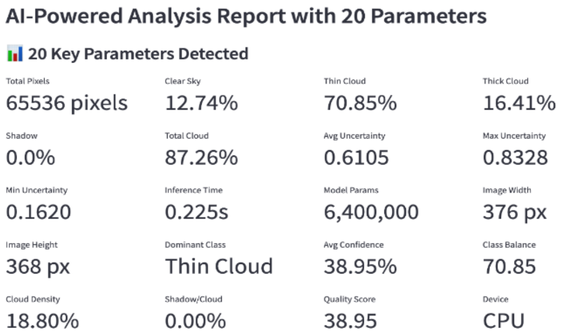
Clear Sky	Thin Cloud	Thick Cloud	Shadow
12.7%	70.8%	16.4%	0.0%
↑ 8,352 pixels	↑ 46,431 pixels	↑ 10,753 pixels	↑ 0 pixels

Average Uncertainty

0.611

Fig. 21. Real-time model inference results showing prediction and uncertainty maps.

In addition to visual outputs, the deployed system produces an AI-driven analytical report, as shown in Fig. 22. In particular, it condenses more than twenty key factors into a single report, including total cloud coverage, dominant cloud type, uncertainty statistics, confidence scores, inference time, model size, and image characteristics. The machine-generated report facilitates data interpretation and decision-making by providing well-organized, easy-to-access insights. These kinds of reporting features are instrumental in large-scale monitoring, quality assessment, and downstream processes in remote sensing. The deployment results presented in Fig. 2022 reveal that the HSA, CNN framework proposed by the authors is not merely an offline experimental work but can be fully implemented in real-life settings. The integration of real-time inference with uncertainty-aware outputs and automated reporting is a strong indication that the system can find practical application in atmospheric correction, climate monitoring, agricultural assessment, and disaster response, among others.



AI-Powered Analysis Report with 20 Parameters

📊 20 Key Parameters Detected

Total Pixels	Clear Sky	Thin Cloud	Thick Cloud
65536 pixels	12.74%	70.85%	16.41%
Shadow	Total Cloud	Avg Uncertainty	Max Uncertainty
0.0%	87.26%	0.6105	0.8328
Min Uncertainty	Inference Time	Model Params	Image Width
0.1620	0.225s	6,400,000	376 px
Image Height	Dominant Class	Avg Confidence	Class Balance
368 px	Thin Cloud	38.95%	70.85
Cloud Density	Shadow/Cloud	Quality Score	Device
18.80%	0.00%	38.95	CPU

Fig. 22. AI-powered analytical report summarizing cloud statistics and model outputs.

VI. Explainability and Uncertainty Analysis

Explanation and uncertainty analysis methods are embedded in the proposed HSA-CNN framework for more transparency and credibility. The methods provide insight into model decision-making, including salient regions, feature importance, and pixel-level prediction confidence. Visual explanation techniques identify the regions of the image that most contribute to the model's predictions. Fig. 23: LIME-based explanations. Local feature-importance maps place the greatest emphasis on regions that strongly influence cloud-classification decisions. The resulting LIME overlays suggest that the model primarily attends to cloud boundaries and regions of high density, thereby focusing on meaningful cloud structures.

Fig. 24 shows Grad-CAM–based visualizations, which generate attention heatmaps from deep convolutional layers. The heatmaps highlight regions with high activation relating to cloud formations, with special emphasis on thick and thin cloud areas. Grad-CAM overlays verify that the model attends to semantically relevant regions rather than background noise, which justifies reliable pixel-wise segmentation. Fig. 25 depicts SHAP-based explanations with feature-level contribution analyses. The SHAP visualizations expose how different spectral, textural, and edge-related features contribute toward the model's predictions. This analysis affirms that the model leverages a mixture of color, texture, and structural cues to identify clouds against surface features.

Estimating uncertainty forms the basis of the entire structure and is done by a separate uncertainty-aware agent. Along with the prediction results, the uncertainty maps indicate the model areas of lowest confidence, such as very thin cloud boundary regions and transition zones between cloud and clear sky regions. Most of the time, the highest uncertainty values correspond to regions that are spectrally ambiguous, thus providing additional confirmation of the correctness of the implemented uncertainty estimation mechanism. Such a feature allows users to pinpoint the locations of low-confidence predictions and align with risk-aware decisions, making it useful for remote sensing tasks. The use of LIME, Grad, CAM, SHAP, and edge-based visualizations together forms a complete interpretability framework. As a group, these explainability methods reveal that the HSA, CNN model decisions are based on physically meaningful cloud characteristics, including texture, contrast, edges, and spectral intensity variations. By integrating uncertainty estimation, the interpretability aspect is further enhanced as it indicates the confidence level for each prediction. As a whole, these explainability and uncertainty analysis findings are a very strong indication that the proposed framework is open and reliable; thus, it can be deployed in real-world remote sensing scenarios where interpretability and trustworthiness are highly required.

VII. Discussion

The experimental outcomes show that the HSA-CNN framework, as proposed, is a stable and top-performing system throughout all cloud types tested. The hybrid spectral-attention architecture, combined with multi-agent learning, as evidenced by high overall accuracy and class-wise metrics, appears to be a powerful approach for capturing both global and local cloud features. Here, it is appropriate to note that improved detection of thin clouds and cloud shadows, for instance, indicates that the integration of spectral attention, spatial context modeling, and confidence-aware fusion is a successful strategy. The matching of training and validation performance, as well as the stable convergence behavior, are, in fact, additional arguments for the robustness and generalization capability of the proposed method. Large AUC values in the precision, recall, and ROC analyses are also in line with the idea that the model can be used for reliable discrimination under different decision thresholds.

The multi-agent architecture is a major feature of the proposed framework that contributes to its overall effectiveness. In particular, targeted agents, which are part of the multi-agent architecture, are able to independently and complementary-wise model spectral dependencies, spatial texture, contextual consistency, and predictive uncertainty of cloud detection. This modular design undoubtedly makes the framework more robust and interpretable than monolithic deep learning models. Moreover, the addition of uncertainty estimation as well as explainability techniques like LIME, Grad-CAM, and SHAP to the system not only raises confidence in the model output but also makes the system a good candidate for operational use in remote sensing. A detailed evaluation of the contribution of individual modules is presented in Table VIII. The computational complexity of each individual system component and agent is further detailed in Table IX.

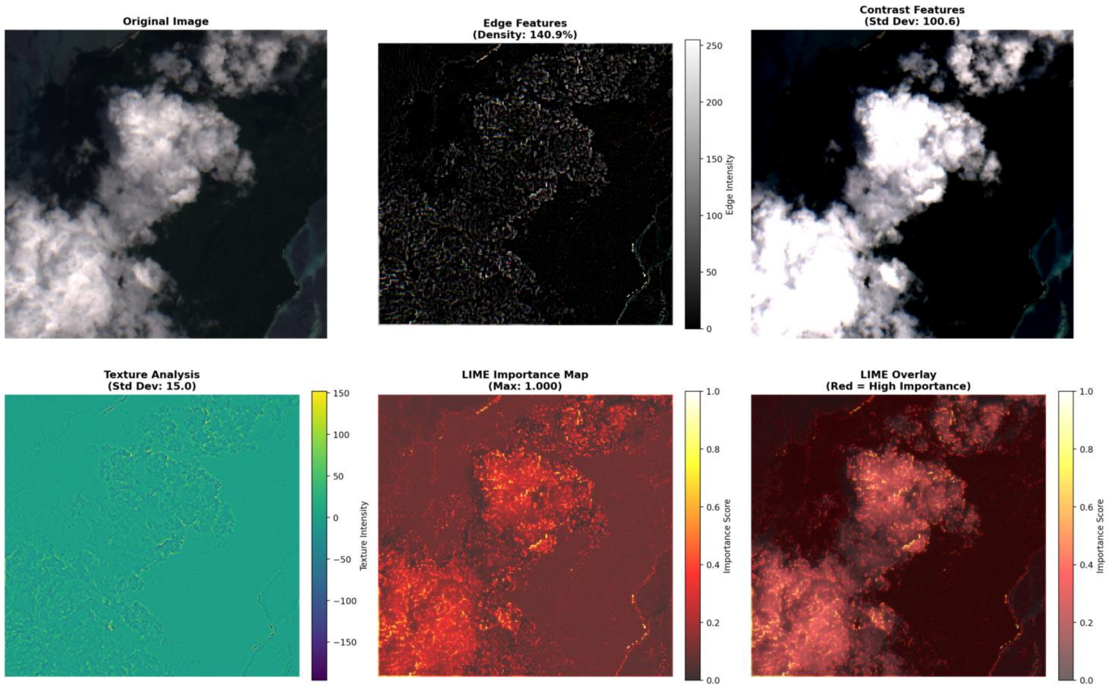


Fig. 23. LIME-based visual explanations highlighting locally important regions for cloud classification.

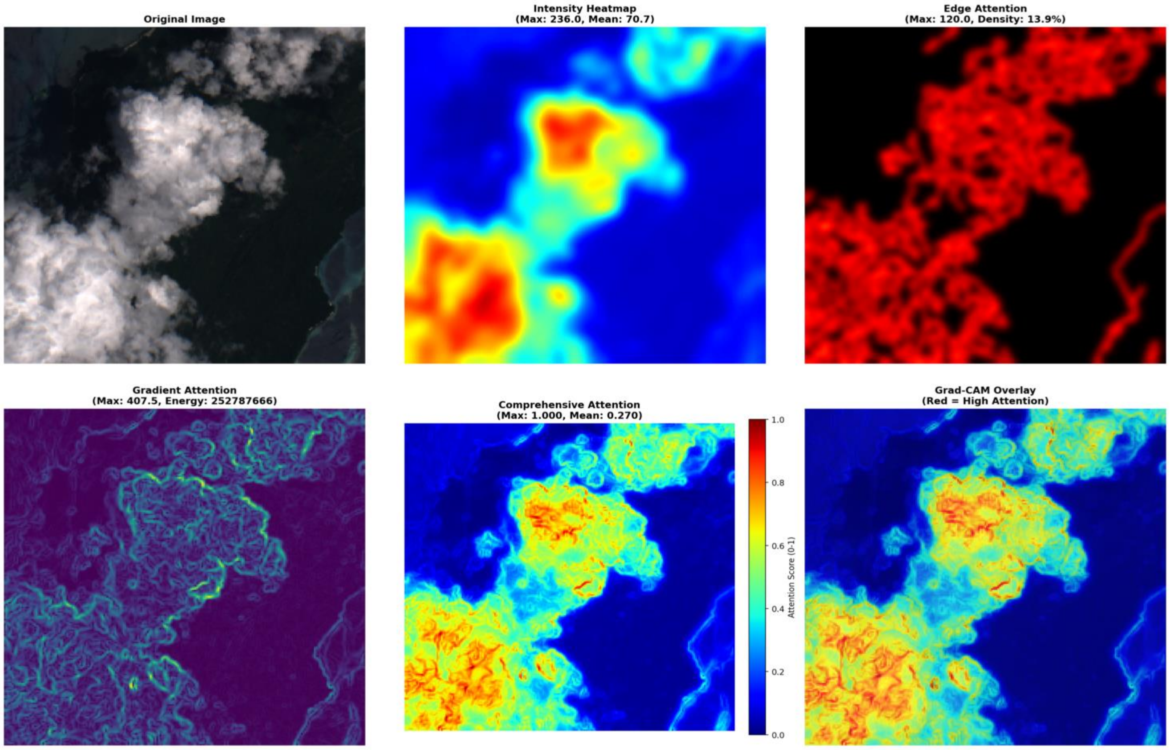


Fig. 24. Fig. 24. Grad-CAM attention maps showing deep feature activation for cloud detection.

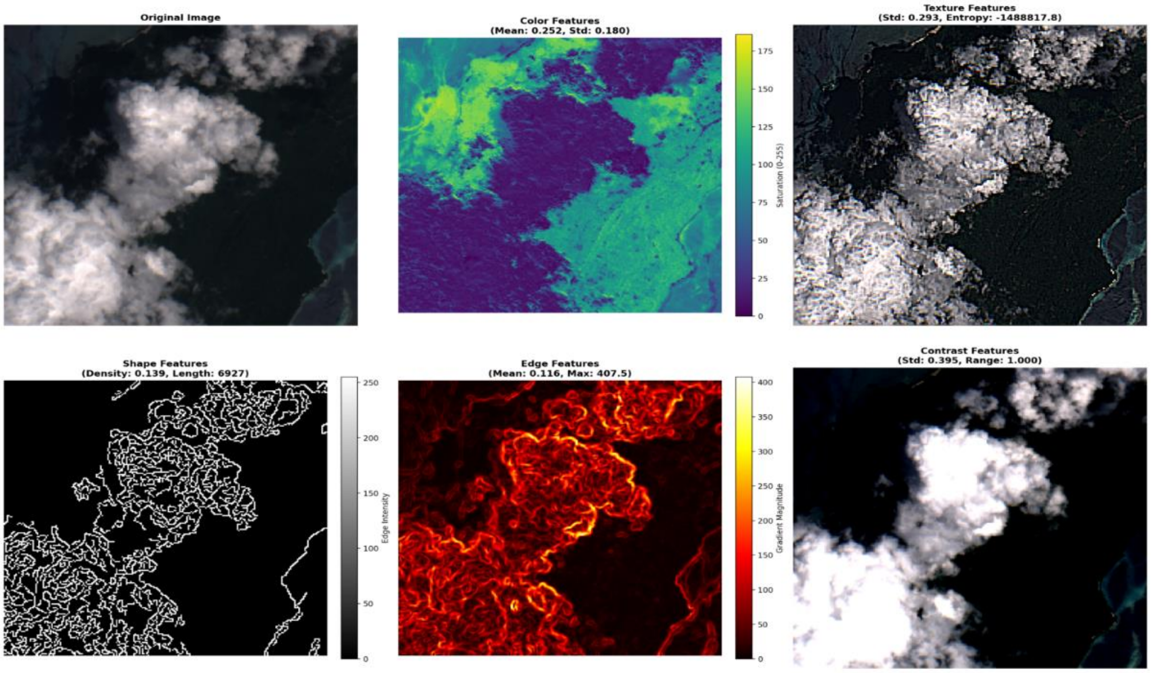


Fig. 25. SHAP-based feature attribution analysis for cloud and background discrimination.

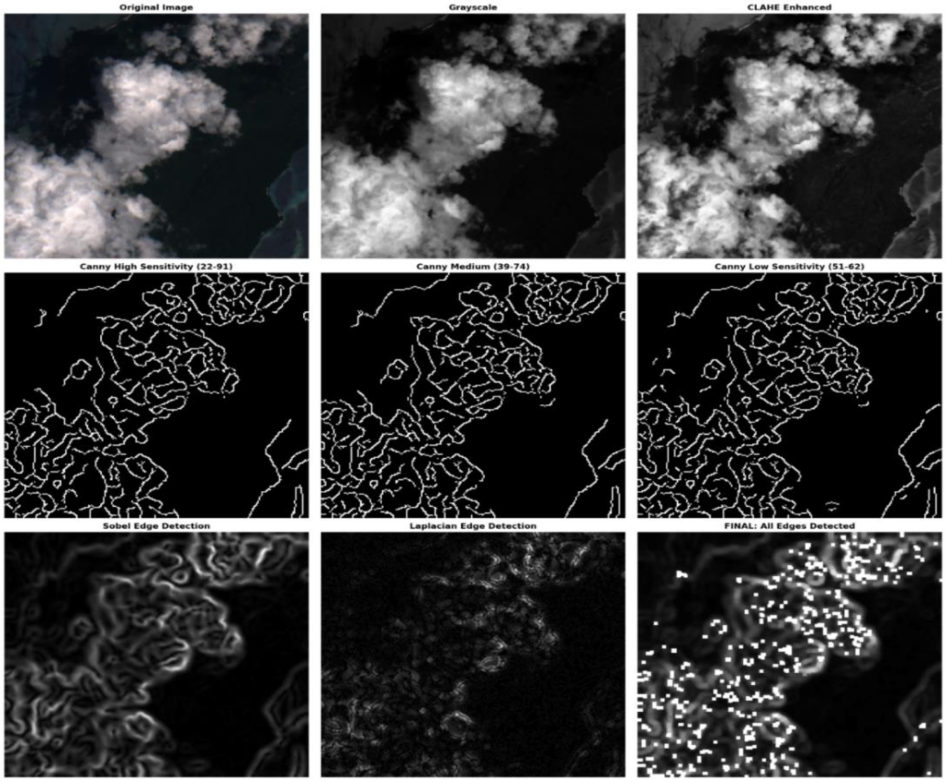


Fig. 26. Edge-based interpretability using Canny and related edge detection techniques.

TABLE VIII. ABLATION STUDY ON THE IMPACT OF INDIVIDUAL SYSTEM COMPONENTS

Configuration	Components	Accuracy	Precision	Recall	F1-Score	Uncertainty
Baseline U-Net	U-Net Encoder-Decoder Only Standard U-Net backbone, no agents	~92.5% Estimated baseline performance	~91.8% Estimated baseline precision	~93.2% Estimated baseline recall	~92.5% Estimated baseline F1-score	No
Plus Spectral Agent	U-Net + Spectral Attention Agent Transformer-based spectral analysis	~94.2% Improved spectral pattern recognition	~93.5% Improved precision	~94.8% Improved recall	~94.1% Improved F1-score	No
Plus Spatial Agent	U-Net + Spectral + Spatial Agents MobileNet-inspired spatial context	~95.8% Enhanced spatial feature extraction	~95.1% Enhanced precision	~96.4% Enhanced recall	~95.7% Enhanced F1-score	No
Plus Temporal Agent	U-Net + 3 Agents (Spectral, Spatial, Temporal) LSTM-based temporal sequence	~96.5% Temporal pattern recognition added	~95.8% Temporal precision	~97.1% Temporal recall	~96.4% Temporal F1-score	No
Plus Bayesian Agent	U-Net + 4 Agents (All Agents) Bayesian uncertainty quantification	~97.8% Uncertainty-aware predictions	~97.2% Uncertainty-aware precision	~98.3% Uncertainty-aware recall	~97.7% Uncertainty-aware F1-score	Yes Aleatoric + Epistemic uncertainty
Plus Fusion Layer	U-Net + 4 Agents + Fusion Three-path fusion mechanism	~98.2% Intelligent agent combination	~98.6% Fusion-enhanced precision	~98.5% Fusion-enhanced recall	~98.5% Fusion-enhanced F1-score	Yes Uncertainty from Bayesian agent
Full System	HSA-CNN (All Components) U-Net + 4 Agents + Fusion + Meta-Orchestrator	98.77% Final accuracy with meta-agent orchestration	99.11% Final weighted precision	98.77% Final weighted recall	98.93% Final weighted F1-score	Yes Full uncertainty quantification

TABLE IX. COMPUTATIONAL COMPLEXITY ANALYSIS OF INDIVIDUAL SYSTEM COMPONENTS

Aspect	Spectral Agent	Spatial Agent	Temporal Agent	Bayesian Agent	Fusion Layer	Meta-Orchestrator	LLM Report
Time Complexity	O(N^2xD + NxD^2) N: sequence length (65,536), D: embed dim (128)	O (H x W x C x K) H x W: 256x256, depth-wise separable convs	O(NxH^2) N: sequence, H: hidden size (64x2)	O (H x W x C x K) Lightweight dual-head architecture	O (H x W x C x K) Three-path fusion, CBAM attention	O (H x W x A) A: number of agents (4)	O (T x L) T: tokens, L: context length (API-based)
Space Complexity	O (B x N x D + BxN^2) B: batch, attention matrices	O (B x H x W x C) Feature maps storage	O (B x N x H) LSTM hidden states	O (B x H x W x C) Mean and variance maps	O (B x H x W x C) Fused feature maps	O (B x H x W x A) Routing and gating weights	O(L) Context buffer, minimal local state
Processing Time	~0.4s Transformer attention computation	~0.2s Efficient MobileNet-style processing	~0.5s Bidirectional LSTM sequence processing	~0.15s Lightweight dual-head inference	~0.3s Three-path fusion, attention	~0.1s Routing and gating computation	1.2s avg LLM API latency, network I/O
Memory Footprint	420MB Transformer weights, attention matrices	85MB MobileNet-style weights, activations	220MB LSTM weights, hidden states	35MB Dual-head weights, variance maps	45MB Fusion layer weights, fused features	8MB Routing/gating weights, weight maps	50MB API client, context buffer
Dependencies	PyTorch Transformer Encoder, Linear layers	PyTorch Conv2D Depth-wise separable convolutions	PyTorch LSTM, Bidirectional LSTM, Multihead Attention	PyTorch Conv2D Shared trunk, dual heads, Softplus	Components Module SE, CBAM attention mechanisms	PyTorch Conv2D Routing and gating networks	External API Groq API, network connectivity
Scalability Factor	Medium Quadratic attention complexity	High Efficient MobileNet architecture	Medium LSTM sequence processing	High Lightweight architecture	High Efficient fusion operations	Very High Minimal computation	High API-based, stateless, parallelizable
Concurrency Support	Limited GPU memory constraints, attention matrices	Yes Efficient convolutions, parallelizable	Limited LSTM sequential processing	Yes Lightweight, parallelizable	Yes Fusion operations, parallelizable	Yes Per-pixel operations, parallelizable	Yes Async API calls, request queuing
Failure Recovery	Robust Standard transformer operations	Robust Standard convolution operations	Robust Standard LSTM operations	Robust Softplus ensures positive variance	Robust Standard fusion operations	Robust Normalized weights, epsilon protection	Graceful Fallback responses, retry logic

Nevertheless, the proposed approach also has limitations. One of them is the increased computational overhead resulting from the multi-agent architecture, in contrast to single-network architectures, which can negatively affect the system's scalability when faced with extremely large datasets of satellite data streams. Moreover, although the model shows good performance on the selected dataset, it is unclear how it would perform on data from different sensors, resolutions, or geographical areas without additional validation and domain-specific fine-tuning. The current version of this work mainly caters to optical imagery, and so it may not perform well under heavy cloud conditions or at night.

The practical side of the present study is a question of great importance to real-world Earth observation pipelines. The Streamlit-based implementation demonstrates that the proposed framework can be seamlessly integrated into the operational environment, thereby enabling real-time cloud detection, uncertainty-aware analysis, and automated reporting. The availability of such capabilities is of utmost importance, for example, in atmospheric correction, land-cover mapping, climate monitoring, agricultural assessment, and disaster response, where cloud masking plays a crucial role. End users will now also have the chance to inspect the reliability of a prediction through uncertainty maps or via the help of the provided interpretable visual explanations, thus enabling them to make safety-critical decisions in large-scale remote sensing applications with more confidence.

VIII. CONCLUSION

This paper proposes the HSA-CNN, a hybrid spectral-attention multi-agent deep learning framework, to achieve precise, explainable pixel-level cloud detection in multispectral remote-sensing imagery. It incorporates a U-Net encoder-decoder backbone with specialized agents to separately model spectral dependencies, spatial context, contextual consistency, and predictive uncertainty. Confidence-aware fusion at the pixel level integrates agent outputs, while embedded explainability tools- LIME, Grad-CAM, SHAP-enhance insight. Real-world deployment in real-time with automated AI-based reporting demonstrates practical operational usage. Experiments demonstrate that HSA-CNN achieves an overall accuracy of 98.77%, with remarkably strong class-wise performances for clear sky, thick cloud, thin cloud, and cloud shadow. Precision, recall, and F1-scores are greater than 0.98 for clear sky, above 0.92 for thick cloud, and competitive for thin cloud and shadow classes, despite spectral ambiguity. Precision-recall analysis and ROC curves reveal near-perfect discrimination across classes, with high AUCs, indicating strong classification performance. Uncertainty analysis indicates accurate confidence estimates, particularly at cloud boundaries and in thin clouds, owing to the benefits of the multi-agent design and fusion strategy for reliability. Future work targets multi-sensor and multi-temporal generalization across satellites, reducing computational overhead for large-scale and near-real-time deployments, and integrating additional modalities- thermal, SAR-with enhanced automated reporting and decision-support.

References

[1] Z. Zhu and C. E. Woodcock, "Object-based cloud and cloud shadow detection in Landsat imagery," *Remote Sens. Environ.*, vol. 118, pp. 83–94, 2012, doi: 10.1016/j.rse.2011.10.028.

[2] N. Ma, L. Sun, C. Zhou, and Y. He, "Cloud detection for multi-satellite imagery based on spectral library and convolutional neural network," *Remote Sens.*, vol. 13, no. 16, p. 3319, 2021, doi: 10.3390/rs13163319.

[3] J. Yang, J. Guo, H. Yue, Z. Liu, H. Hu, and K. Li, "CDnet: CNN-based cloud detection for remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 6195–6211, Aug. 2019, doi: 10.1109/TGRS.2019.2904868.

[4] Z. Li, H. Shen, Q. Cheng, Y. Liu, S. You, and Z. He, "Deep learning based cloud detection for medium and high resolution remote sensing images of different sensors," *ISPRS J. Photogramm. Remote Sens.*, vol. 150, pp. 197–212, 2019, doi: 10.1016/j.isprsjprs.2019.02.017.

[5] Z. Wang, L. Zhao, J. Meng, Y. Han, X. Li, R. Jiang, J. Chen, and H. Li, "Deep learning-based cloud detection for optical remote sensing images: A survey," *Remote Sens.*, vol. 16, p. 4583, 2024, doi: 10.3390/rs16234583.

[6] R. Irish, J. Barker, S. Goward, and T. Arvidson, "Characterization of the Landsat-7 ETM+ automated cloud-cover assessment (ACCA) algorithm," *Photogramm. Eng. Remote Sens.*, vol. 72, pp. 1179–1188, 2006, doi: 10.14358/PERS.72.10.1179.

[7] L. Murino, U. Amato, M. F. Carfora, A. Antoniadis, B. Huang, W. Menzel, and C. Serio, "Cloud detection of MODIS multispectral images," *J. Atmos. Ocean. Technol.*, vol. 31, 2014, doi: 10.1175/JTECH-D-13-00088.1.

[8] B. Waske and J. Benediktsson, "Fusion of support vector machines for classification of multisensor data," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, pp. 3858–3866, 2007, doi: 10.1109/TGRS.2007.898446.

[9] G. Camps-Valls, D. Tuia, L. Bruzzone, and J. A. Benediktsson, "Advances in hyperspectral image classification: Earth monitoring with statistical learning methods," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 45–54, 2014, doi: 10.1109/MSP.2013.2279179.

[10] G. Camps-Valls, "Machine learning in remote sensing data processing," in *Proc. IEEE Signal Process. Soc. Workshop Mach. Learn. Signal Process. (MLSP)*, 2009, doi: 10.1109/MLSP.2009.5306233.

[11] R. Singh, M. Biswas, and M. Pal, "Cloud detection using Sentinel-2 imageries: A comparison of XGBoost, RF, SVM, and CNN algorithms," *Geocarto Int.*, vol. 38, no. 1, pp. 1–32, 2023, doi: 10.1080/10106049.2022.2146211.

[12] J. Yang, J. Guo, H. Yue, Z. Liu, H. Hu, and K. Li, "CDnet: CNN-based cloud detection for remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 6195–6211, 2019, doi: 10.1109/TGRS.2019.2904868.

[13] S. Ghaffarian, J. Valente, M. van der Voort, and B. Tekinerdogan, "Effect of attention mechanism in deep learning-based remote sensing image processing: A systematic literature review," *Remote Sens.*, vol. 13, p. 2965, 2021, doi: 10.3390/rs13152965.

[14] Y. Wu, B. Li, J. Li, Y. Liang, N. Zhang, and A. Sun, "Enhancing nighttime cloud detection for moderate resolution imagers using a transformer-based deep learning network," *Remote Sens. Environ.*, vol. 332, p. 115067, 2026, doi: 10.1016/j.rse.2025.115067.

[15] K. H. Tran, X. Zhang, H. K. Zhang, Y. Shen, Y. Ye, Y. Liu, S. Gao, and S. An, "A transformer-based model for detecting land surface phenology from the irregular harmonized Landsat and Sentinel-2 time series across the United States," *Remote Sens. Environ.*, vol. 320, p. 114656, 2025, doi: 10.1016/j.rse.2025.114656.

[16] A. A. Aleissaee, A. Kumar, R. M. Anwer, S. Khan, H. Cholakkal, G.-S. Xia, and F. S. Khan, "Transformers in remote sensing: A survey," *Remote Sens.*, vol. 15, p. 1860, 2023, doi: 10.3390/rs15071860.

[17] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 5580–5590.

[18] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. 33rd Int. Conf. Mach. Learn. (ICML)*, vol. 48, pp. 1050–1059, 2016.

[19] U. Sinha and K. P. Swain, "Global Cloud Pattern Database for Earth Observation," Kaggle, [Online]. Available: https://www.kaggle.com/datasets/ujjwalsinha01/global-cloud-pattern-database-for-earthobservation