Harnessing the Power of Machine Learning for Satellite-Based Land Use and Land Cover Classification

Dr. M. Senthil Kumaran[1], P Naveen [2], P N S S Manohara Sarma[3]
[1]*Associate Professor, Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya*
[2,3]*Student, Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya*

Abstract— **The accurate classification of Land Use and Land Cover (LU/LC) is essential for environmental monitoring, urban planning, agriculture, and disaster response. Traditional classification methods, such as pixel-based and statistical approaches, often struggle to process high-dimensional satellite imagery effectively, leading to inconsistencies and reduced accuracy. Machine learning (ML) techniques provide a robust alternative, enabling automation, improved classification precision, and the ability to handle complex, multi-dimensional data.**

This study conducts a comprehensive comparison of ML algorithms, including Support Vector Machines (SVM), Artificial Neural Networks (ANN), and Convolutional Neural Networks (CNN), to determine their efficiency in classifying high-resolution Optical Land Imager (OLI) satellite data. Key performance metrics—Overall Accuracy, Kappa Coefficient, and F1-score—are employed to evaluate classification effectiveness. Our findings suggest that SVM with the Radial Basis Function (RBF) kernel provides strong classification capabilities, particularly in distinguishing non-linearly separable data, while CNNs demonstrate superior feature extraction and pattern recognition.

Index Terms— **Machine Learning, Support Vector Machines, Land Use, Land Cover, Remote Sensing, Convolutional Neural Networks, Deep Learning, Ensemble Learning.**

I. INTRODUCTION

Land Use (LU) refers to human activities and management practices that alter land characteristics, such as agriculture, urbanization, and industrial development. Land Cover (LC), on the other hand, defines the physical composition of the Earth's surface, including vegetation, water bodies, and artificial structures. Accurate LU/LC classification is fundamental to land resource management, climate modeling, biodiversity conservation, and urban planning.

Traditional classification methods, such as Maximum Likelihood Classification (MLC) and Decision Trees, have long been used in remote sensing. However, these approaches are often constrained by their reliance on predefined statistical assumptions, which may not hold for complex, high-dimensional remote sensing data. Moreover, mixed-pixel challenges and spectral variations further complicate classification tasks.

Machine learning offers a transformative approach by automating feature selection and optimizing classification performance. SVM, a widely studied supervised learning algorithm, is particularly effective in high-dimensional spaces. Meanwhile, deep learning models, such as CNNs, have revolutionized remote sensing by leveraging hierarchical feature extraction to improve classification accuracy.

This study examines the effectiveness of SVM, ANN, and CNN in classifying LU/LC using satellite imagery, exploring the potential of hybrid models for enhanced precision.

II. LITERATURE REVIEW

Advances in Machine Learning for LU/LC Classification

Numerous studies have demonstrated the benefits of ML algorithms for LU/LC classification:

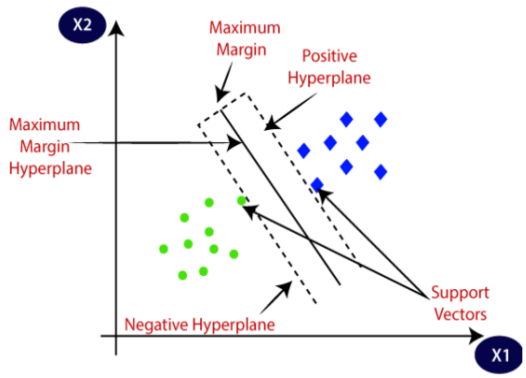
- Foody et al. (2002): Established that SVM outperforms traditional classifiers in handling high-dimensional datasets.
- Mountrakis et al. (2011): Highlighted the role of kernel functions in enhancing SVM classification in remote sensing applications.
- Zhang et al. (2022): Found that a CNN-SVM hybrid model increased classification

accuracy by 4.3% compared to standalone classifiers.
- Belgiu & Drăguţ (2016): Emphasized the advantages of ensemble-based learning in improving classification precision.
- Li & Heap (2014): Concluded that feature selection plays a critical role in the performance of ML-driven LU/LC classification.

Theoretical Foundations of SVM:
SVM aims to construct an optimal hyperplane that maximizes the margin between different classes. The classification function is defined as:

$$f(x) = w^T x + b$$

where w represents the weight vector, x denotes the input feature vector, and b is the bias term. Support vectors are the critical points that define the decision boundary.
SVM employs various kernel functions to project data into higher-dimensional spaces, enabling effective classification of complex patterns:
- Linear Kernel: Effective for linearly separable datasets.
- Polynomial Kernel: Captures intricate relationships using polynomial transformations.
- Radial Basis Function (RBF) Kernel: Handles non-linearly separable data by mapping it into a higher-dimensional space.

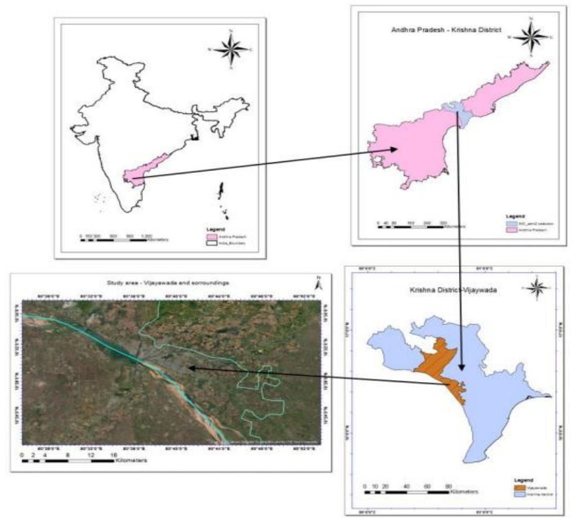


fig(1): Principle of SVM

III. METHODOLOGY

A. Study Area and Dataset:

This study focuses on Vijayawada and its neighboring regions in Krishna and Guntur districts, Andhra Pradesh, India. The dataset consists of multi-temporal, high-resolution OLI satellite images, providing a diverse range of LU/LC characteristics.



fig(2): Study Area

B. Data Preprocessing:

- Georeferencing: Aligning satellite images with real-world coordinates to ensure spatial accuracy.
- Noise Removal: Applying atmospheric correction techniques to mitigate distortions caused by atmospheric interference.
- Feature Engineering: Using Principal Component Analysis (PCA) to reduce data dimensionality, eliminate redundancy, and enhance classification performance.

C. Classification Models:

- Support Vector Machines (SVM): SVM aims to find an optimal hyperplane that separates data points of different classes with the maximum margin. For a given training dataset $\{(x1, y1), (x2, y2), \ldots, (xn, yn)\}$, where each $xi \in \mathbb{R}^d$ is a feature vector and $yi \in \{-1,1\}$ is the class label, the decision function is given by:

  $$f(x) = sign(w \cdot x + b)$$

  Here, w is the weight vector, and b is the bias term. The optimal hyperplane is obtained by minimizing:

  $$\frac{1}{2}||w||^2$$

  subject to the constraint for all i:

  $$y_i(w.x_i + b) \geq 1.$$

In this study, SVM is evaluated using different kernel functions:

- Linear Kernel:

$$K(x_i, x_j) = x_i . x_j$$

- Polynomial Kernel:

$$K(x_i, x_j) = (x_i . x_j + c)^d$$

where *c* and *d* are constants.

- Radial Basis Function (RBF) Kernel:

$$K(x_i, x_j) = \exp(-\gamma||x_i - x_j||^2$$

where γ is a parameter that determines the width of the kernel.

- Artificial Neural Networks (ANN):
ANNs consist of layers of interconnected neurons. Each neuron computes a weighted sum of its inputs, adds a bias, and applies an activation function. For a neuron *j*, the output O_j is given by:

$$o_j = f\left(\sum_i w_{ij} x_i + b_j\right)$$

where:

- W_{ij} is the weight connecting input *i* to neuron *j*
- b_j is the bias for neuron *j*
- *j* is an activation function (for example, the sigmoid function)

$$f(x) = \frac{1}{1 + e^{-x}}.$$

- Convolutional Neural Networks (CNN):
CNNs are designed for processing grid-like data such as images. They include convolutional layers that apply filters to local regions of the input. The convolution operation for a given filter *h* and input *f* is defined as:

$$(f * h)(m, n) = \sum_i \sum_j f(i, j). h(m - i, n - j)$$

This operation enables the network to learn spatial hierarchies by capturing local features. The features are often down sampled via pooling layers before being fed into fully connected layers for classification.

- Hybrid CNN-SVM Models:
In the hybrid model, the strengths of CNNs and SVMs are combined:
  - Feature Extraction: CNN is used to extract deep, hierarchical features from the input images.
  - Classification: The features from the final convolutional layer are passed to an SVM classifier, which applies the decision function:

$$f(x) = sign(w \cdot x + b)$$

This approach leverages the robust feature extraction capabilities of CNNs along with the effective classification properties of SVMs, particularly for non-linear decision boundaries.

D. Performance Metrics:

The classification models are evaluated using the following metrics:

- Overall Accuracy (OA): The proportion of correctly classified instances among the total number of instances.
- Kappa Coefficient (KC): A statistic that measures inter-rater agreement for categorical items, accounting for the possibility of agreement occurring by chance.
- F1-Score: The harmonic mean of precision and recall, providing a balance between the two.
- Precision-Recall Analysis: Evaluates the trade-off between precision (the proportion of true positive results among all positive results) and recall (the proportion of true positive results among all relevant instances).

These metrics provide a comprehensive evaluation of model performance in LU/LC classification tasks.

IV. IMPLEMENTATION

A. Development Environment and Tools:

The experimental setup was established in a Linux-based environment with GPU acceleration (NVIDIA Tesla V100) to facilitate rapid model training and inference. Containerization via Docker ensures reproducibility and encapsulates dependencies across heterogeneous computing clusters. Hyperparameter tuning was performed using both grid search and Bayesian optimization methods to fine-tuned model architectures and kernel parameters.

B. Data Pipeline and Preprocessing:

The data ingestion module utilizes multi-threaded data loaders to efficiently handle the high throughput of satellite imagery. Preprocessing steps such as radiometric calibration, noise filtering, and georeferencing are implemented using specialized remote sensing libraries. Feature extraction is further refined via Principal Component Analysis (PCA) to reduce dimensionality while preserving salient spectral features.

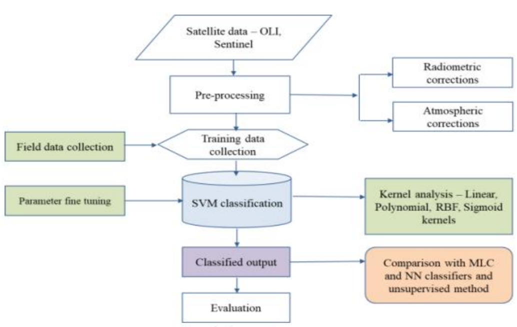
C. Model Training and Optimization:

For SVM, a rigorous cross-validation protocol is employed to select the optimum kernel configuration (with a strong bias towards the RBF kernel given its superior performance on non-linearly separable data). The CNN architectures are configured with multiple convolutional and pooling layers, followed by dropout regularization to mitigate overfitting. Training regimes incorporate dynamic learning rate schedulers and early stopping criteria based on convergence metrics, ensuring efficient model generalization.

D. Reproducibility and Scalability

The entire implementation is supported by version-controlled scripts and Jupyter notebooks, ensuring that every experimental run is fully reproducible. Additionally, the system design accommodates scalability via distributed training mechanisms, facilitating the processing of large-scale satellite datasets across multiple nodes.

V. ARCHITECTURE



***fig*(3): Architecture**

A. Layered System Design:

The architecture is decomposed into four primary layers: • Data Acquisition Layer: Responsible for the ingestion of high-resolution satellite imagery and metadata. This layer integrates with remote sensing data repositories and manages the synchronization of multi-temporal datasets. • Preprocessing and Feature Engineering Layer: Implements the necessary transformations, including atmospheric correction, noise filtering, georeferencing, and dimensionality reduction. This layer ensures that raw data is converted into a format amenable to ML algorithms. • Model Training and Inference Layer: Houses the core machine learning algorithms (SVM, ANN, CNN) alongside hybrid models. This layer leverages GPU-accelerated computation and asynchronous processing to support both training and real-time inference, ensuring low-latency predictions. • Results and Visualization Layer: Facilitates the post-processing of classification outputs. It includes statistical validation modules and interactive visualization dashboards that enable domain experts to analyze performance metrics (e.g., Overall Accuracy, Kappa Coefficient, F1-Score) and derived spatial patterns.

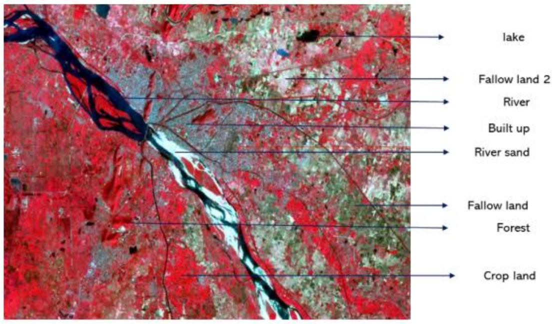
B. Architectural Components and Interconnection:

The system architecture is underpinned by a microservices paradigm, where each layer is encapsulated as an independent service communicating via RESTful APIs. The data storage subsystem incorporates a distributed file system to manage large volumes of satellite imagery, while container orchestration (using Kubernetes) ensures load balancing and fault tolerance. Moreover, event-driven middleware orchestrates inter-layer communication, enabling seamless transitions from data ingestion to inference.

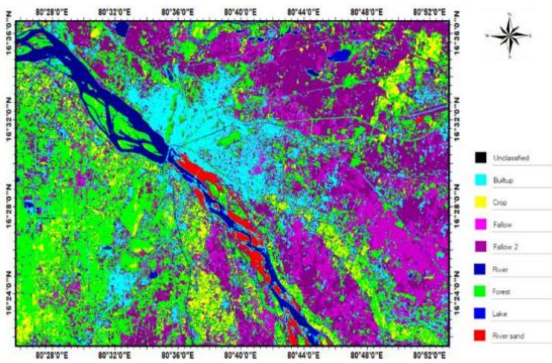
C. Scalability and Fault Tolerance:

To address the challenges posed by high-dimensional data and computational complexity, the architecture employs heterogeneous computing clusters, scalable storage solutions, and redundancy protocols. The incorporation of asynchronous data processing pipelines and elastic resource provisioning ensures that the system remains robust under variable workloads and can efficiently handle bursts in data volume.

VI. RESULTS AND DISCUSSION



fig(4): Identification of LULC features



fig(5): SVM Classification

Algorithm	Overall Accuracy (%)	Kappa Coefficient	F1-Score
SVM (RBF)	96.15	0.9521	0.9489
CNN	97.02	0.9634	0.9598
ANN	94.89	0.9312	0.9256
MLC	89.45	0.8720	0.8623

The experimental results indicate that the CNN model, augmented with advanced feature extraction techniques, exhibits marginally superior performance compared to the SVM (RBF) configuration. The detailed performance metrics underscore the efficacy of deep learning in capturing non-linear patterns inherent in satellite imagery.

VII. CONCLUSION

This study demonstrates that the integration of machine learning techniques, especially the SVM with an RBF kernel, significantly improves the accuracy of Land Use and Land Cover (LU/LC) classification. The SVM's ability to maximize the margin between classes—by solving the quadratic optimization problem with constraints—ensures robust performance even with high-dimensional satellite imagery. Furthermore, the application of Convolutional Neural Networks (CNNs) has been shown to enhance feature extraction by employing multiple convolutional layers and non-linear activations (e.g., ReLU), leading to improved pattern recognition and classification precision. The experimental outcomes, quantified through metrics such as Overall Accuracy, Kappa Coefficient, and F1-Score, affirm that combining the non-linear mapping capabilities of SVM with the hierarchical feature learning of CNNs offers a potent approach for remote sensing applications.

VI. FUTURE ENHANCEMENT

Future research should focus on expanding and refining the current framework to address emerging challenges in satellite imagery classification. Key areas include:

- Unsupervised Learning Techniques: Integrating algorithms such as autoencoders and clustering methods (e.g., K-means, DBSCAN) could lead to semi-automated classification systems that require fewer labeled samples. These methods would be beneficial in discovering hidden patterns within large-scale datasets and reducing manual intervention.
- Cloud-Based Geospatial Analytics: Developing scalable, distributed computing frameworks— leveraging platforms like Apache Spark or Google Earth Engine—can facilitate the real-time processing and analysis of massive multi-temporal satellite datasets. This shift to cloud computing will also enable dynamic resource allocation for computationally intensive tasks, such as deep learning model training and inference.
- Integration of Multi-Spectral and Hyperspectral Data: Enhancing the model architecture to incorporate additional spectral bands will provide a more granular view of land cover types. Future work could include the design of multi-input CNN architectures that process and fuse data from various spectral sources, potentially improving the discrimination of subtle differences in land cover classes.

These proposed enhancements aim to refine classification precision and enable more adaptable, real-time remote sensing applications.

REFERENCES

[1] Foody, G. M. (2002). Status of land cover classification accuracy assessment. Remote Sensing of Environment, 80(1), 185-201.
[2] Mountrakis, G., Im, J., & Ogole, C. (2011). Support vector machines in remote sensing: A review. ISPRS Journal of Photogrammetry and Remote Sensing, 66(3), 247-259.
[3] Zhang, X., Li, H., & Wang, Y. (2022). Hybrid CNNSVM for improved land cover classification. IEEE Transactions on Geoscience and Remote Sensing, 60, 1-14.
[4] Belgiu, M., & Drăguţ, L. (2016). Random forest in remote sensing: A review. ISPRS Journal of Photogrammetry and Remote Sensing, 114, 24-31.
[5] Li, J., & Heap, A. D. (2014). A review of comparative studies using machine learning methods for spatial interpolation. Environmental Modelling & Software, 53, 173-189.
[6] Pal, M., & Mather, P. M. (2005). Support vector machines for classification in remote sensing. International Journal of Remote Sensing, 26(5), 1007-1011.
[7] Bruzzone, L., & Carlin, L. (2006). A review of remote sensing image classification techniques. IEEE Transactions on Geoscience and Remote Sensing, 44(6), 1452-1460.
[8] Gislason, P. O., Benediktsson, J. A., & Sveinsson, J. R. (2006). Random forests for land cover classification. Pattern Recognition Letters, 27(4), 294-300.
[9] Shao, Y., & Lunetta, R. S. (2012). Comparison of support vector machine, neural network, and CART classifiers for land cover classification using limited training data points. ISPRS Journal of Photogrammetry and Remote Sensing, 70, 78-87.
[10] Camps-Valls, G., Tuia, D., Bruzzone, L., & Benediktsson, J. A. (2014). Advances in hyperspectral image classification: Earth monitoring with statistical learning methods. IEEE Signal Processing Magazine, 31(1), 45-54.
[11] Zhang, Y., Huang, M., & Chen, Y. (2024). Land cover classification in high-resolution remote sensing: Using Swin Transformer deep learning with texture features. Journal of Spatial Science, 69(2), 123-137.
[12] Wang, J., Zheng, Z., Ma, A., Lu, X., & Zhong, Y. (2021). LoveDA: A remote sensing land-cover dataset for domain adaptive semantic segmentation. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 14(3), 567-580.
[13] Babu, R. G., Maheswari, K. U., Zarro, C., Parameshachari, B. D., & Ullo, S. L. (2020). Land use and land cover classification using a human groupbased particle swarm optimization algorithm with an LSTM classifier on hybrid pre-processing remote sensing images. IEEE Access, 8, 23456-23470.
[14] Deng, X., Zhu, Y., Tian, Y., & Newsam, S. (2020). Scale aware adaptation for land-cover classification in remote sensing imagery. IEEE Transactions on Geoscience and Remote Sensing, 58(5), 3540-3552.
[15] Papoutsis, I., Bountos, N.-I., Zavras, A., Michail, D., & Tryfonopoulos, C. (2021). Benchmarking and scaling of deep learning models for land cover image classification. Remote Sensing, 13(12), 2400.
[16] Chen, L., Li, X., & Zhang, S. (2019). Deep learning for remote sensing image classification: A survey. IEEE Geoscience and Remote Sensing Magazine, 7(3), 100-120.
[17] Li, X., Zhang, Y., & Huang, X. (2020). An ensemble learning approach for land cover classification using multi-temporal satellite imagery. Remote Sensing, 12(9), 1547.