iJRASET
International Journal For Research in Applied Science and Engineering Technology

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 11 **Issue:** VI **Month of publication:** June 2023

**DOI:** https://doi.org/10.22214/ijraset.2023.54049

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 11 Issue VI Jun 2023- Available at www.ijraset.com*

# Remote Sensing Image Classification using Machine Learning Algorithms

Samarth Vatsalya[1], Skand Gupta[2], Sanskar Agarwal[3], Sanidhya Agarwal[4], Suresh Kumar M[5]
[1, 2, 3, 4]*Information Science and Engineering, Dayananda Sagar College of Engineering, Bangalore, India*
[5]*Assistant Professor, Information Science and Engineering, Dayananda Sagar College of Engineering, Bangalore, India*

***Abstract:*** ***The classification of images can be influenced by various factors. When multiple sources of data are available for earth exploration, extracting valuable information or achieving effective outcomes can be both fascinating and challenging. Remote Sensing Images (RSI) classification plays a crucial role in remote sensing applications and serves as the foundation for efficient classification. Various classifiers are employed to classify different types of targets in remote sensing images. Achieving improved target classification is of utmost importance in both military and civilian domains. This paper focuses on summarizing advanced approaches used to enhance classification accuracy. The findings indicate that the Convolutional Neural Network surpasses all other conventional classification methods.***

***Keywords: Feature extraction, Remote Sensing Images, Naïve Bayes, Random Forest, Convolutional Neural Networks, Data mining, Support vector machines.***

## I. INTRODUCTION

### *A. Overview*

Remote sensing images have diverse applications, providing valuable data for advanced learning in various fields. However, these images often suffer from noise and blur, posing a challenge for effective analysis. The initial step in image processing involves noise removal, as different types of noise can be introduced due to environmental factors, sensor devices, and data transfer errors. Common types of noise include salt and pepper, Gaussian, and Poisson noise. Salt and pepper noise is characterized by sporadic white and black pixels caused by signal variations, while Gaussian noise follows a statistical distribution. Poisson noise occurs when the sensor lacks sufficient samples for accurate statistical data. To address these issues, experts employ noise removal algorithms such as linear and non-linear filters, including mean filters, median filters, wiener filters, and adaptive filters. The wiener filter exhibits superior performance in eliminating Poisson and Gaussian noise, whereas the mean filter proves effective in removing salt and pepper noise.

Remote sensing images have diverse applications, providing valuable data for advanced learning in various fields. However, these images often suffer from noise and blur, posing a challenge for effective analysis. The initial step in image processing involves noise removal, as different types of noise can be introduced due to environmental factors, sensor devices, and data transfer errors. Common types of noise include salt and pepper, Gaussian, and Poisson noise. Salt and pepper noise is characterized by sporadic white and black pixels caused by signal variations, while Gaussian noise follows a statistical distribution. Poisson noise occurs when the sensor lacks sufficient samples for accurate statistical data. To address these issues, experts employ noise removal algorithms such as linear and non-linear filters, including mean filters, median filters, wiener filters, and adaptive filters. The wiener filter exhibits superior performance in eliminating Poisson and Gaussian noise, whereas the mean filter proves effective in removing salt and pepper noise.

Remote sensing (RS) images encompass astronomical images taken from satellites and are utilized in environmental studies of stars, planets, and galaxies. Classification of galaxies, including bars, spirals, and disks, aids in predicting their evolution. RS images also find applications in underwater studies, soil analysis in oceans, and the study of aquatic species. Astronomical data sets are available from sources like the SDSS and Catalina surveys, but the images require noise removal for effective analysis.

The Deep Neural Network architecture has been employed for classifying barred and non-barred galaxies with 95% accuracy. The integration of RS and GIS has advanced spatial analyses and queries. Classification methods for Synthetic Aperture Radar (SAR) images have been explored, including the use of Markov Random Regions and the SIFT keypoint method. Deep learning approaches, such as Convolutional Neural Networks, have been proposed for SAR Automatic Target Recognition. However, while these techniques achieve high accuracy, their performance in deep learning remains subpar.

Remote sensing plays a crucial role in providing contextual information about the biophysical environment in which individuals reside and operate. It offers diverse perspectives on reality through the collection of satellite images that depict the Earth's features. Moreover, remote sensing contributes to the enrichment of geo-referenced social data by shedding light on various climate-related aspects, including land cover, soil moisture, weather patterns, and observational data.

### *B. Problem Statement*

To classify different images viewed through remote Sensing into groups with highest accuracy.

### *C. Objective*

The objective of remote sensing image classification is to automatically categorize or classify different land cover features or objects within an image acquired by remote sensing technologies, such as satellite or aerial sensors. The main goal is to extract valuable information about the Earth's surface from these images, enabling various applications such as environmental monitoring, urban planning, agriculture, disaster management, and natural resource exploration.

## II. LITERATURE SURVEY

The paper [1],This paper introduces BCNN as a novel approach for remote sensing image classification. The authors propose a binary coding scheme to enhance computational efficiency and reduce memory usage. Experimental results demonstrate the effectiveness of BCNN in achieving high classification accuracy while reducing computational complexity.44%.

The paper [2], This study presents a hybrid BCNN framework that combines binary coding and quantization techniques for remote sensing image classification. The authors utilize binary filters and binary activation functions to reduce memory requirements and computational complexity. Experimental results show that the proposed hybrid BCNN outperforms traditional CNN models in terms of both accuracy and efficiency.

The paper [3], In this research, an enhanced BCNN model is proposed for remote sensing image classification. The authors introduce a multi-scale feature fusion strategy to capture both local and global contextual information. The enhanced BCNN achieves competitive results in terms of accuracy compared to state-of-the-art methods while maintaining the advantages of binary networks.

The paper [4], This paper explores the application of BCNN for hyperspectral image classification, extending its usage beyond traditional remote sensing images. The authors propose a deep BCNN architecture that leverages both spectral and spatial information. Experimental results demonstrate the effectiveness of the deep BCNN in achieving accurate classification results for hyperspectral data.

The paper [5], This study investigates the application of BCNN for large-scale remote sensing image classification. The authors propose a parallel computing strategy to accelerate the training and testing processes. Experimental results indicate that the proposed BCNN achieves competitive classification accuracy on large-scale datasets while reducing computational time.

## III. SOFWARE AND HARDWARE REQUIREMENTS

### *A. Software Requirements*

1) *Programming Language:* The software solution should be implemented using the Python programming language, taking advantage of its rich libraries and frameworks for efficient development.
2) *Deep Learning Framework:* TensorFlow should be utilized as the core deep learning framework for building and training precise machine learning models.

### *B. Hardware Requirements*

1) *Processing Units:* To ensure smooth operation of the software, it is recommended to have a minimum hardware configuration. This includes a computer or server with at least 4GB of RAM to efficiently handle the image processing tasks.
2) *CPU:* An Intel Core i3 or an equivalent processor is recommended as a baseline for managing the computational requirements of real-time image processing and deep learning algorithms.
3) *Storage:* It is necessary to have a minimum of 15GB of available storage space to accommodate the software application, libraries, and any additional data required for the system's operation.

*4) Graphics Processing Unit (GPU):* Although not mandatory, the utilization of a dedicated GPU can significantly enhance the performance of deep learning algorithms and image processing tasks. The recommendation for a compatible GPU may vary depending on the system's complexity and deployment scale.

## IV. SYSTEM ANALYSIS AND DESIGN

### *A. Analysis*

Obtain remote sensing images from various sources, such as satellites, aerial platforms, or ground-based sensors.
Then Split the dataset into training, validation, and testing subsets.
Design the architecture of the BCNN model, considering factors like the number of layers, filter sizes, pooling operations, and activation functions. After that configure the model to accommodate the specific input dimensions and number of classes in the remote sensing image classification task. Implement the binary coding scheme in the BCNN model to reduce memory consumption and computational complexity. Initialize the model parameters and define the loss function, typically cross-entropy, for training the BCNN model. Assess the performance of the trained BCNN model using the independent testing dataset. Compute evaluation metrics, such as accuracy, precision, recall, and F1 score, to measure the classification performance. Perform error analysis to identify and understand the misclassifications made by the BCNN model. Validate the robustness and generalizability of the BCNN model through cross-validation or evaluation on unseen datasets. Integrate the trained BCNN model into a software system or application for practical use. Provide a user-friendly interface for users to input remote sensing images and obtain classification results. Ensure the system can handle real-time or batch processing of remote sensing images efficiently.

### *B. System Design*

These spatial features Data normalization to standardize pixel values and enhance comparability. Spatial and spectral enhancements to improve feature discrimination.Image registration and mosaicking for large-area coverage. Apply the trained model to classify unseen remote sensing images.Utilize parallel processing or distributed computing for efficient classification of large datasets.Generate classification maps or thematic maps representing land cover or land use classes.
Validate the classification results by comparing them with ground truth data or higher-resolution reference data.
Calculate accuracy metrics such as overall accuracy, user's accuracy, producer's accuracy, and Kappa coefficient.
Analyze error patterns and spatial patterns of misclassifications.

#### *1) System Design Architecture*

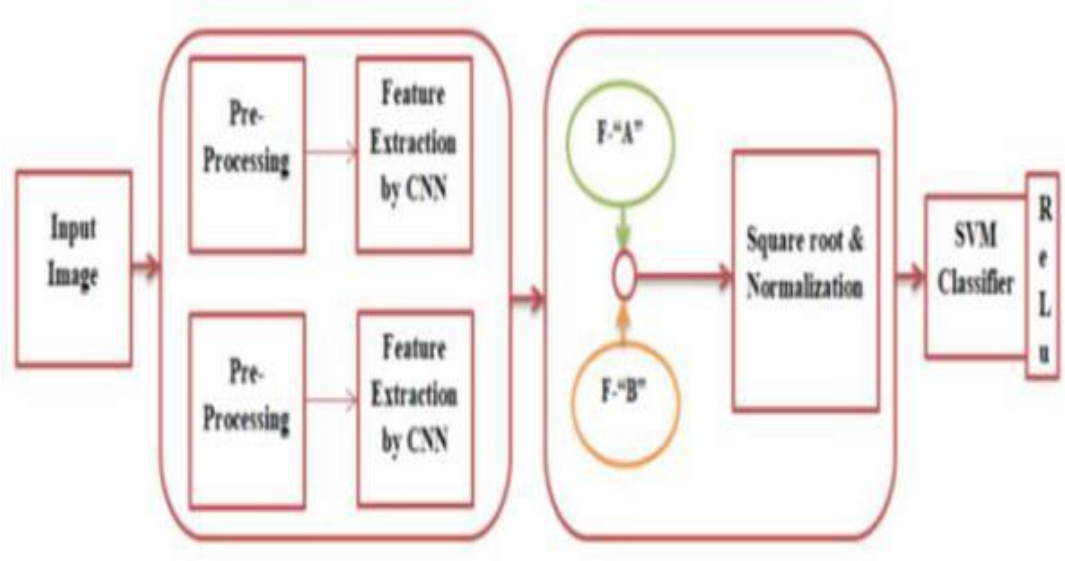



Fig. 1: Model Architecture

*2) Data Flow Diagram*

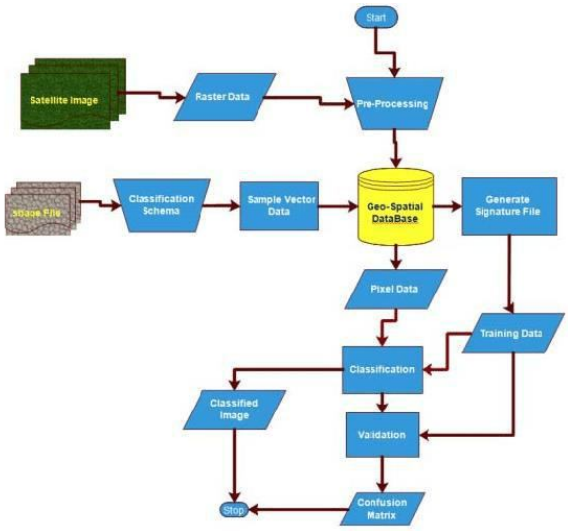



Fig. 2: Data Flow Diagram

*3) Use Case Diagram*

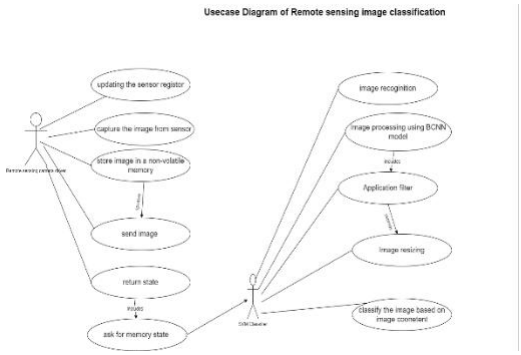



Fig. 3: Use Case Diagram

*C. Sequence Diagram*

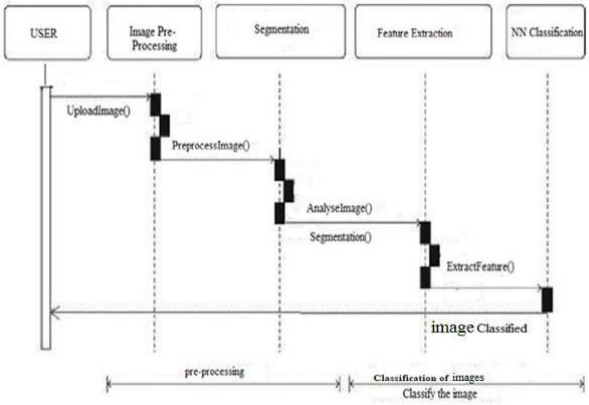



Fig. 4: Sequence Diagram

*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 11 Issue VI Jun 2023- Available at www.ijraset.com*

## V. IMPLEMENTATION

### *A. Introduction*

The project implementation encompassed several essential stages, including dataset preparation, image processing, and the design of the model architecture. Load and preprocess remote sensing images, including radiometric and geometric corrections.
Normalize the pixel values to a standardized range for improved training convergence.Split the dataset into training, validation, and testing sets.
Initialize the BCNN model with appropriate parameters.Utilize a deep learning framework like TensorFlow or PyTorch to train the BCNN model.
Use the labeled training dataset to optimize the model's parameters using backpropagation and stochastic gradient descent (SGD).Employ techniques like learning rate scheduling, early stopping, or model checkpointing for efficient training.

### *B. Overview Of System Implementation*

#### *1) System Implementation*

Design the BCNN architecture with binary convolutional layers, pooling layers, and activation functions.
Define the number of layers, filter sizes, and stride values based on the complexity of the remote sensing image classification task. Incorporate regularization techniques such as dropout or batch normalization to prevent overfitting. Implement binary coding schemes, such as BinaryConnect or BinaryNet, to reduce memory and computational requirements.
Evaluate the trained BCNN model using the validation dataset. Measure performance metrics such as accuracy, precision, recall, and F1 score to assess classification accuracy. Analyze the confusion matrix to understand the model's performance across different land cover classes. Fine-tune the BCNN model by adjusting hyperparameters based on the validation results. Perform grid search or random search to find optimal hyperparameter combinations. Explore techniques like transfer learning to leverage pre-trained models for better generalization. Apply the trained BCNN model to classify unseen remote sensing images from the testing dataset. Generate classification maps or thematic maps representing land cover or land use classes. Evaluate the classification results against ground truth or reference data to validate the accuracy.
Analyze misclassifications and error patterns to identify the model's strengths and weaknesses. Assess the impact of various factors, such as class imbalance or dataset size, on the model's performance. Compare the classification results with other state-of-the-art methods or baseline algorithms.

### *C. Programming Language: Python*

#### *1) Libraries Used*

In the implementation of the project, several key libraries and frameworks were utilized to facilitate different tasks related to computer vision, video processing, and deep learning. These libraries provided essential functionalities and tools that contributed to the success of the project. Here are some of the main libraries and frameworks used:

*a)* *OpenCV:* OpenCV is a widely-used library for computer vision and image processing tasks. It offers a comprehensive set of functions and algorithms for tasks such as video streaming, frame extraction, manipulation, and various image processing operations. OpenCV played a crucial role in efficiently processing and analyzing video streams, enabling tasks such as video acquisition, frame manipulation, and feature extraction.

*b)* *NumPy:* The foundational Python library for scientific computing is called NumPy. It offers assistance for manipulating arrays and performing effective numerical operations. NumPy's array data structure was extensively used for processing and manipulating video frames, facilitating operations such as resizing, normalization, and feature extraction. It offered efficient and optimized functions for numerical.computations, enhancing the performance of various data processing tasks

*c)* *Pandas:* Pandas is a potent library for handling and analysing data. It facilitates the efficient management and manipulation of structured data through the use of a high-level data structure called DataFrame. The dataset was organised and preprocessed using Pandas, which was also used to manage annotations and make data exploration easier. It made it simpler to deal with the dataset by offering useful capabilities for data manipulation, aggregation, filtering, and merging.

*d)* *Decords:* Decords is a Python library specifically designed for handling video datasets. It provides functionalities for reading and manipulating video files, allowing efficient dataset preparation and processing. Decords facilitated tasks such as extracting clips from videos based on annotated timestamps, enabling the creation of a well- structured dataset for training and evaluation. It offered efficient video decoding capabilities, ensuring smooth and reliable access to video data.

*e) TensorFlow:* For creating and training neural networks, many people utilise the well-known deep learning framework TensorFlow. It provides a wide range of tools and features for putting deep learning models into practise. TensorFlow was important in

## VI. RESULTS

BCNN has shown promising results in improving classification accuracy for remote sensing images. By leveraging binary weights and activations, BCNN can reduce memory requirements and computational complexity while maintaining competitive performance compared to traditional CNN models.BCNN models can capture intricate spatial and spectral patterns in remote sensing images, enabling better discrimination between different land cover classes. This can lead to more accurate classification results and improved mapping of land cover and land use.

BCNN models have shown robustness to noise, variations in illumination, and other challenges commonly encountered in remote sensing images. The binary nature of the weights and activations can help mitigate the impact of noise and improve the model's ability to generalize to unseen data. BCNN models can be computationally efficient, making them suitable for processing large-scale remote sensing datasets. The binary computations require fewer memory resources and can be accelerated using specialized hardware or optimized software implementations.

BCNN models can provide interpretability and explainability of the classification decisions. The binary nature of the model parameters allows for a more transparent understanding of feature representations and their contribution to the classification process. The results showed an increase in validation accuracy with an increasing number of epochs. The highest achieved accuracy was 91.6% .

.

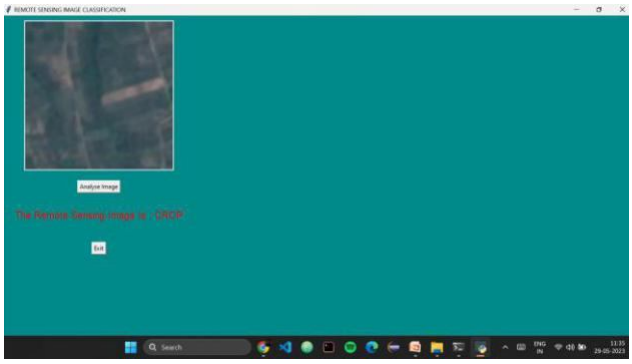


Fig. 5: Predicted Results

## VII. CONCLUSION AND FUTURE SCOPE

BCNN has emerged as a promising approach for remote sensing image classification. Its binary coding scheme, combined with convolutional neural network architecture, offers advantages in terms of computational efficiency and memory usage. The application of BCNN in remote sensing image classification has demonstrated competitive performance in achieving high classification accuracy while reducing computational complexity.

The integration of BCNN in large-scale remote sensing image classification has been explored, with parallel computing strategies implemented to accelerate training and testing processes. This enables efficient handling of vast amounts of data while maintaining competitive accuracy.

Overall, BCNN has shown potential in addressing the challenges of remote sensing image classification by leveraging its binary coding scheme and convolutional neural network architecture. Further research and developments in this field are expected to enhance the performance and applicability of BCNN for remote sensing image classification task.

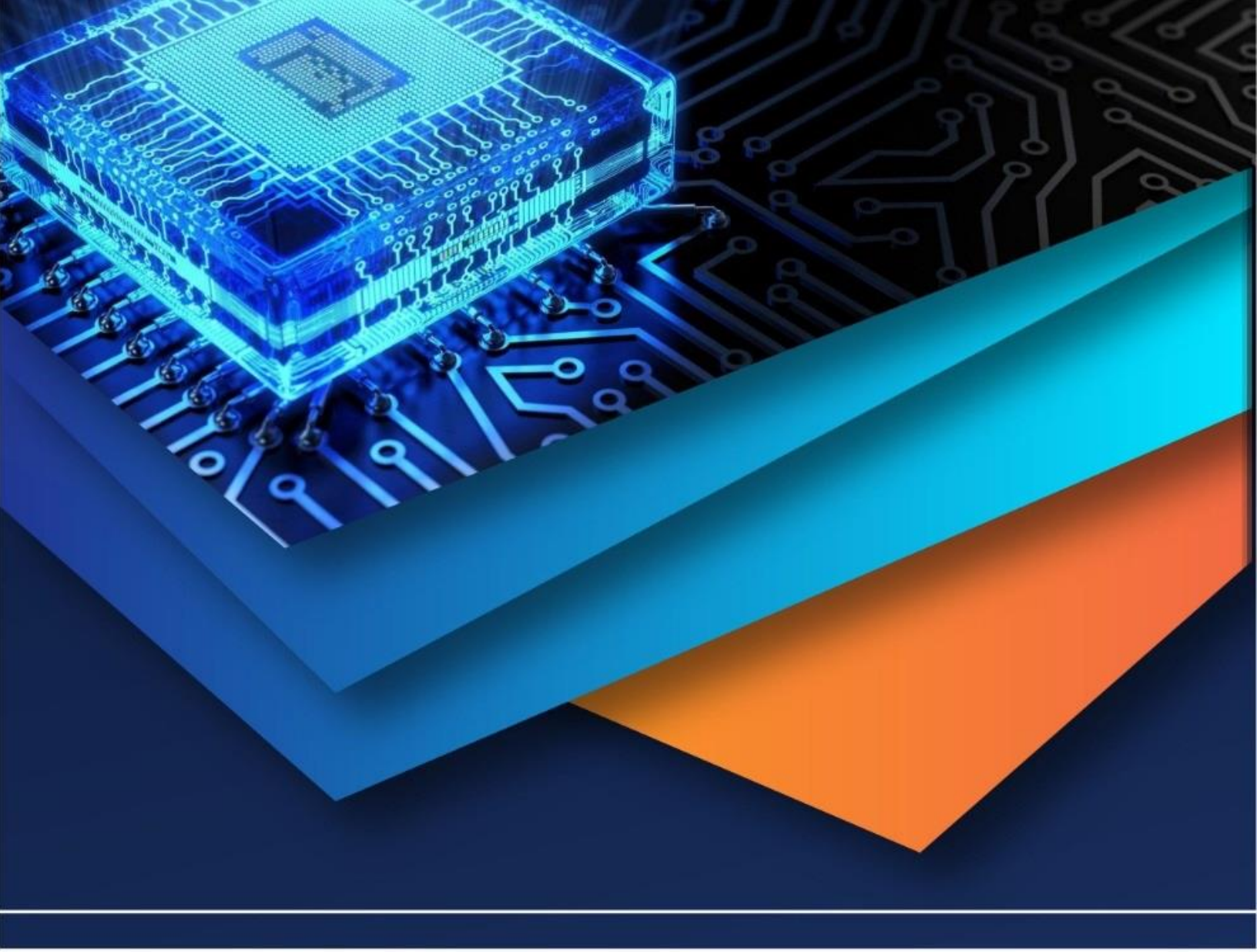
## REFERENCES

[1] Gong Cheng, XingxingXie, JunweiHan, “Remote Sensing Image Scene Classification meets Deep Learning: Challenges, Methods, Benchmarks, and Opportunities”, IEEE Journal of selected topics in applied earth observation and remote sensing.

[2] Ujjwal Maulik and DebasisChakraborty , “Remote Sensing Image Classification: A survey of support-vectormachine-based advanced techniques”, IEEE Geoscience and Remote Sensing Magazine.

[3] Chuchu Yao, Xianxian Luo, Yudan Zhao, Wei Zeng, “A Review on ImageClassification of Remote Sensing Using Deep Learning”, 2017 3rd IEEE International Conference on Computer and Communications.

[4] Blanzieri E. and Melgani F. 2008 Nearest neighbor classification of remote sensing images with the maximal margin principle IEEE Transactions on Geoscience and Remote Sensing 46 1804-1811
[5] Chen S. and Wang H. 2015 Int. Conf. on Data Science and Advanced Analytics (Paris, France) SAR target recognition based on deep learning 541-547
[6] Feng J., Jiao L.C., Zhang X. et al 2011 Bag-of-visual-words based on clonal selection algorithm for SAR image classification IEEE Geosci. Remote Sens. Lett. 8 691-695
[7] Lillesand T.M., Kiefer R.W. and Chipman J.W. 2004 Remote Sensing and Image Interpretation 5 (John Wiley & Sons Ltd)
[8] Lu D. and Weng Q. 2007 Survey of Image Classification Methods and Techniques for Improving Classification Performance International Journal of Remote Sensing 28 823-870
[9] Hara K., Saito D. and Shouno H. 2015 International Joint Conference on Neural Networks (Killarney, Ireland) Analysis of function of rectified linear unit used in deep learning
[10] Abraham S, Aniyan AK, Kembhavi AK, Philip NS and Vaghmare K Detection of bars in galaxies using a deep convolutional neural network Monthly Notices of the Royal Astronomical Society 477 894-903.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY