International Journal of Basic and Applied Sciences, 14 (SI-2) (2025) 93-99

International Journal of Basic and Applied Sciences

Website: www.sciencepubco.com/index.php/IJBAS
https://doi.org/10.14419/j6er8175
Research paper

GLCM-ResNet: Deep Neural Model for Noise Removal and Multiband Image Classification

Vaibhav J. Babrekar *, Shirish M. Deshmukh

Department of Electronics & Telecommunication, Prof Ram Meghe Institute of Technology & Research, Amravati, India
**Corresponding author E-mail:vaibhav.babrekar@gmail.com*

Received: April 30, 2025, Accepted: July 3, 2025, Published: July 18, 2025

Abstract

Hyperspectral image classification plays a crucial role in various remote sensing applications, requiring deep learning models that offer both high accuracy and stability. In this study, we propose “CGLM-tweaked ResNet-16”, an optimized variant of ResNet-16, demonstrating superior performance across hyperspectral datasets. Our experiments on the “Indian Pines” and “Pavia University” datasets reveal that, CGLM ResNet-16 outperforms standard ResNet-16, particularly in terms of accuracy and loss reduction. For the Indian Pines dataset, CGLM ResNet-16 achieves an impressive 99.88% accuracy with the lowest loss of 2.8%, surpassing other competing models. Similarly, for the Pavia University dataset, the model maintains low loss 0.23% while achieving competitive accuracy, signifying improved efficiency and model stability. The reduced loss values indicate better generalization and robustness, crucial for real-world applications. While the pro-posed model enhances classification performance, certain challenges persist, particularly in noise reduction across multiple layers. Future research should explore hybrid deep learning architectures to further optimize accuracy without increasing computational overhead. The biggest challenge ahead is cross domain analysis which remains a critical bottleneck in multiband image processing. Effective noise removal techniques tailored for hyperspectral imaging must be developed to enhance the model’s generalization across diverse datasets. Addressing these challenges will significantly improve real-world applications, such as remote sensing, land cover classification, and environmental monitoring. In conclusion, CGLM ResNet-16 presents a promising statistical analysis method advancement in hyperspectral image classifi-cation, offering improved accuracy and loss minimization.

***Keywords**: GLCM, RESNET, Google Collab, Noise removal, HSI*

1. Introduction

The process of making life better, simpler and healthier is the way of the future. This progress in human mankind has been possible because of an abundance of data which is captured and stored in various ways and means for processing. Power of data in this world of artificial intelligence is the key of moving forward in this fast life where basic information exchange is also ways of detecting human parameters in terms of emotions, biological vitals and many more in and around. One of the key fields of progress with processing needs for this power is satellite imagery. The power comes from the process of capturing images remotely of the atmospheric distant land surfaces with the help of satellites that orbit the planet. These satellite images known as hyperspectral images are captured with the help of multi spectral sensors and camera boarded on the capturing devices that have various layers of parameters captured and stored for processing. This processing provides us great insights required to shape our future in the fields of environment monitoring, urban development and planning, military operations and most importantly agricultural growth. This growth requires extensive power in terms of computations that not only provides end results but also conquers effectiveness, as each image captured is hindered with one or the other form of noise that is embedded in the source capturing device. Noise can be filtered but making noise zero in any captured image is difficult in terms of microscopic parameters like that of sensor limitations, atmospheric challenges and environmental interdependencies [1]. This noise can be random that is asym-metrical in nature or symmetric in nature that is similar patterns and distortions can be induced that can be filtered or removed easily with the help of statistical toolsets applied.
Statistical toolsets not only help in filtering noise but in terms of artificial intelligence they help the node computing in terms of decision making. This specification in terms of machine learning algorithm has key challenges in terms of object detection and classification. Var-ious statistical toolsets that have been utilize to remove noise from images when we talk about computer vision, medical imaging, remote sensing and many more. On the overview the methods can be summarized with the help of table 1 below. Every method has its approach complexity as well as essential utility, but every method has various computational restriction which can be understood in terms of ad-vantages and disadvantages from table 1.

Table 1: Summary of Statistical Methods for Filtering Images

Toolset	Statistical Basis	Advantages	Disadvantages	Literature Citation
Gaussian Filtering	Assumes Gaussian-distributed noise.	Simple, fast, effective for Gaussian noise.	Blurs edges and details.	[2]
Median Filtering	Uses order statistics to remove outliers.	Preserves edges, effective for salt-and-pepper noise.	Less effective for Gaussian noise.	[3]
Wiener Filtering	Assumes noise and image have different statistical properties.	Adaptively adjusts to local noise.	Requires estimation of local variance.	[4]
Bilateral Filtering	Combines spatial and intensity-based Gaussian filtering.	Edge-preserving, effective for Gaussian noise and details.	Computationally expensive.	[5]
Non-Local Means (NLM)	Assumes similar patches across an image share information.	Excellent for preserving details, reduces noise effectively.	Computationally expensive.	[6]
Wavelet Transform	Assumes noise is in high-frequency components.	Effective across multiple resolutions.	Can introduce artifacts if thresholding is too aggressive.	[7]
Total Variation Denoising	Uses differential equations to smooth images.	Preserves edges while reducing noise.	May result in overly smooth images or artifacts.	[8]
Deep Learning (Autoencoders, CNNs)	Learns representations from noisy/clean image datasets.	Excellent for complex noise types and detail preservation.	Requires large datasets and high computational resources.	[9]
Principal Component Analysis (PCA)	Assumes noise is in lower principal components.	Good for high-dimensional noise reduction.	Can lose significant image details if misapplied.	[10]
Anisotropic Diffusion	Diffusion adapts based on local image structure.	Edge-preserving, effective for noise reduction.	May not perform well with very noisy images.	[11]
Kalman Filtering	Models signal and noise as random processes.	Effective for time-varying signals (e.g., video).	Requires good modelling of signal dynamics.	[12]
Markov Random Field (MRF)	Model spatial correlations between neighbouring pixels.	Effective for structured noise (textures, patterns).	Computationally expensive.	[13]
Spatio-Temporal Filtering	Combines spatial smoothing with temporal changes.	Effective for video noise reduction.	Requires handling motion and temporal changes.	[14]

In hyper spectral imaging the main constraint is of spatial relationship of a layer of image pixels with the other layer in terms of neural layer computations. Hence in image processing hyper spectral methodology that can be best utilized is the essence of the paper that can be better understood by methodology in next section.

2. Methodology

In image processing the relationship of one pixel with another especially in spatial domain is the basis of image classification technique which is utilized in any developed algorithm model [15] [16]. Taking this basic requirement into account, the methodology of our algorithm is to utilize the Gray Level Co-occurrence Matrix (GLCM) which is a focused statistical method, for developing our neural network. The need of GLCM is that it provides us with texture and pattern analysis that is beneficial for image classification. These benefits provide added advantage of feature extraction in neural training which can be understood mathematically as below. In terms of basic constraints GLCM is nothing but the formation of a square matrix where the elements of the matrix represent the relationship of one pixel with another that may be in terms of horizontal, vertical or diagonal distance as shown in figure 1 for illustration below:

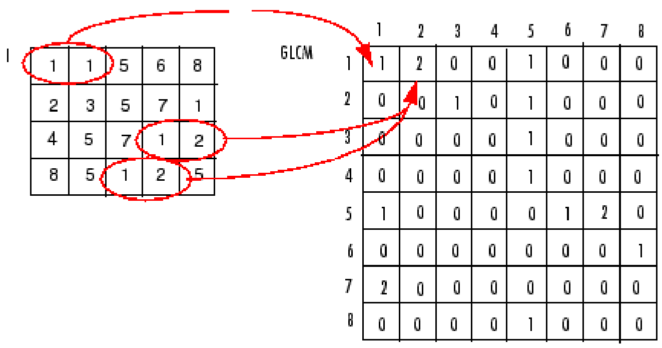


Fig. 1: GLCM matrix formation example

From the above figure the GLCM matrix element 1 is formulated from the original image matrix. The image contains two horizontal adjacent pixels which are both of same value hence the horizontal distance is same. In the next occurrence of the horizontal distance elements 1 and 2 repeat twice hence the second element of GLCM is marked as two as shown. Like this the complete input image is scanned and the GLCM matrix is formulated. So, steps for processing GLCM in terms of algorithm needs are as below:

1. Convert the image to grayscale (if not already done).
2. Choose the spatial relationship: Define the direction and distance for pixel pairs, such as:

Direction: 0°, 45°, 90°, 135° (horizontal, diagonal, vertical directions).

Distance: A specified number of pixels (usually 1 but could be larger).

3. Create the co-occurrence matrix: For each pixel pair in the specified direction and distance, count how often the pair of gray levels appears in the image.

4. Normalize the GLCM: Normalize the matrix by dividing each element by the total number of pixel pairs to get probabilities.

So, methodology comprises of GLCM as shown in the architecture of model in figure 2; this model is not just for the mathematical ease of various attributes above but also for:

1. Feature Extraction from Clustered Patterns - To gather diverse information about satellite images, which can later be used for classification or recognition tasks.
2. Feature Reduction - To streamline the data for more efficient and accurate classification.
3. Ensemble Classification - To identify the optimal classification method that provides the best accuracy for denoised satellite images.
4. Comparison and Validation - To achieve accurate and efficient object recognition in satellite images, enhancing their usability for real-world applications like urban planning, agriculture monitoring, or disaster management.

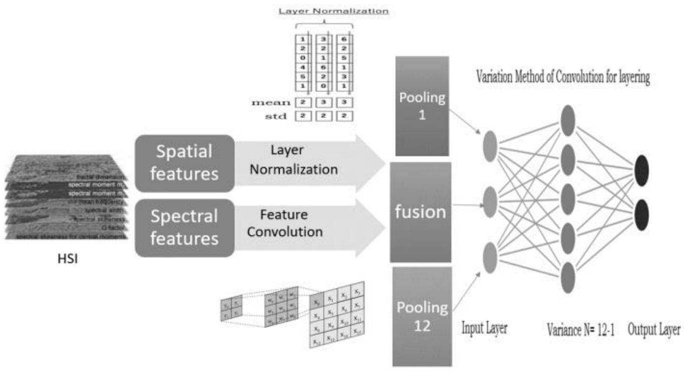


Fig. 2: Architecture of Neural Network

On our research work in the incorporation of GLCM along with the neural network for algorithm development is covered in the next section. To judge the GLCM based methodology the following constraints formulate the means of judging out our developed algorithm as stated in the upcoming section.

- Contrast: Measures the intensity contrast between a pixel and its neighbor.

Contrast = $\sum_{i,j}(i-j)^2 P(i,j)$

- Correlation: Measures how correlated a pixel is to its neighbor.

Correlation = $\frac{\Sigma_{i,j}(i-\mu_x)(j-\mu_y)P(i,j)}{\sigma_x \sigma_y}$

- Energy: Measures the uniformity or smoothness of the image.

Energy = $\sum_{i,j} P(i,j)^2$

- Homogeneity: Measures how uniform the pixel pairs are, with lower values indicating more variability.

Homogeneity = $\sum_{i,j} \frac{P(i,j)}{1+|i-j|}$

- ASM (Angular Second Moment): Like Energy, it measures the uniformity of the GLCM.

ASM = $\sum_{i_1 j} P(i,j)^2$

- Entropy: Measures the disorder or randomness in the texture.

Entropy = $-\sum_{i,j} P(i,j)\log\big(P(i,j)\big)$

- Dissimilarity: Measures the contrast between a pixel and its neighbor.

Dissimilarity = $\sum_{i,j}|i-j|P(i,j)$

3. Algorithm

The research objectives of utilizing the advantages of RESNET in comparison to previous literature reporting [1] requires following spatial needs to be exploited namely feature extraction, dimensionality reduction, and ensemble classification into a pipeline.

Steps of computational pipeline are:

1. **Preprocessing satellite images** – the goal is not only to reduce noise but the requirements of preserving image details for better classification in our algorithm we use median filtering over other reported image segmentation methods for the prime enlisted advantages:

- Noise Reduction – as median filtering cleans images before adding more algorithm computational complexities.
- Edge preservation – the goal of retaining spatial details is achieved with the help of median filtering
- Better Outcomes – This is possible in terms of better results as compared to image segmentation methods of clustering, thresholding, or region-growing techniques, which are sensitive to noise. This is highlighted with our result section along with efficiency as below.
- Efficiency – clean and meaningful data for processing.

2. **Feature extraction with RESNET model** – In this algorithm step we combine the advantages of GLCM statistical computational along with RESNET to bring about a better tweaked GLCM enhanced RESNET16 model which can be renamed as GLCM-RESNET16 or GR16. The name is secondary on compared to the advantages of this model in comparison with standard RESNET.

Why GLCM?

Texture Recognition: Key role in classification.

- **Improved Accuracy**: Higher classification accuracy especially in terms of spatial information.
- **Better Generalization**: Variation Handling.
- **Interpretability**: decision making improvement in terms of classification
- **Multi-modal Fusion**: richer set of features that can improve performance in diverse applications.

3. **Build Classification dataset** - Developing the training dataset.

4. **Evaluation and Implementation** – Comparison of Classifiers in terms of python computations.

➢ **Implementation flow steps are:**

Step Description
1. Load and preprocess the satellite images
2. Pass images through ResNet to extract feature embeddings
3. Compute additional features (GLCM, color, object) from intermediate ResNet layers
4. Reduce feature dimensionality
5. Train and evaluate multiple classifiers, then implement ensemble learning
6. Compare ensemble classification accuracy for optimal object recognition and denoising performance.

➢ **Implementation Requirements**

1. Hardware: Ensure a GPU is available for training ResNet efficiently.
2. Dataset: Use a diverse satellite image dataset with labelled classes (e.g., water, vegetation, urban).
3. Hyperparameter Tuning: Optimize ResNet, feature reduction, and ensemble model parameters.
4. Visualization: Plot confusion matrices, feature importance, and accuracy comparisons.

In our implementation we have chosen python based personal computer specifications over Google Collab because of:

- **No Internet Dependency** – You can run code offline without requiring an internet connection.
- **Full Control Over Environment** – You can configure Python, install specific package versions, and customize settings.
- **Better for Large Datasets** – No restrictions on RAM, disk space, or execution time like Collab's free tier.
- **Supports GUI Applications** – You can build and run GUI-based apps (e.g., PyQt, Tkinter) easily.
- **Easier File Access** – No need to upload/download files; you have direct access to local storage.
- **No Time Limits** – Collab free-tier sessions may disconnect after inactivity or long execution times.

Moreover, node computations in the world of IOT and the need of comparing previous literature citations with our developed model we chose python over Google Collab. The implementation can be visualized better in terms of screenshots as highlighted in the next section of results.

4. Results

The algorithm developed is implemented on a personal computer as per the above-mentioned hardware requirements. The toolchain utilized is "Anaconda Navigator Spyder" where python version utilized is "3.7.16". The training datasets under test are University of Pavia and Indiana Pines for classification of land map. The Indiana Pines and University of Pavia datasets are two commonly used hyperspectral image datasets for remote sensing and classification research. A reported comparison is as highlighted below:
The Indiana Pines and University of Pavia datasets are two commonly used hyperspectral image datasets for remote sensing and classification research. Here's a comparison that's available on Wikipedia:

Table 2: Comparison of Indian Pines and University of Pavia datasets

Feature	Indiana Pines	University of Pavia
Source	AVIRIS sensor (NASA)	ROSIS sensor (ESA)
Location	Northwest Indiana, USA	Pavia, Italy
Spatial Resolution	20 meters per pixel	1.3 meters per pixel
Spectral Bands	220 bands (reduced to 200 after removal of noisy bands)	115 bands
Size (pixels)	145 × 145	610 × 340

Number of Classes	16 (agriculture & forest land cover types)	9 (urban land cover types)
Total Samples	21,025 labelled pixels	42,776 labelled pixels
Key Characteristics	Mixed vegetation and farmland	Urban environment with buildings, roads, etc.

The impact of spatial resolution is to be studied for removal of noise hence two very distinct datasets are chosen as highlighted from above table. One data set is based on agricultural and forestry classification where the region to be classified is very distinctly differentiable whereas other is urban landscape wherein detailed smaller sections need classification. So, the impact of noise in the finer details is studied by our developed algorithm. The study is highlighted with following screenshots for ease of discussion and findings.

Figure 2 high lights that our developed algorithm is implemented using python on the Spyder framework offered by anaconda.

Figure 3 highlights that the developed algorithm provided two means of analyzing the output. One the result provides us with text output as well as console output of the processed image as classified in terms of color. Text output is the essence of statistical measuring constraints, and the console output is the means of visualizing the classified image. the show the successful implementation of our developed algorithm in python. The text file written as shown in below screenshot (figure 4) comprises of all the iterations that are recorded in terms of training and testing.

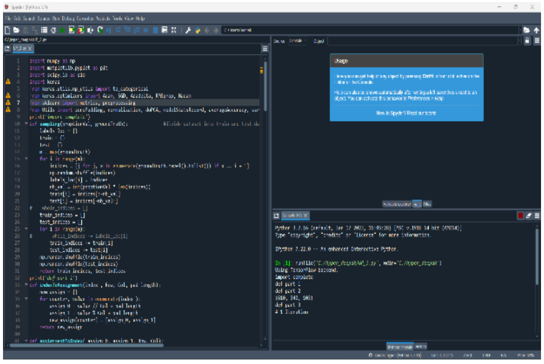


Fig. 2: Screenshot of Python based developed Algorithm

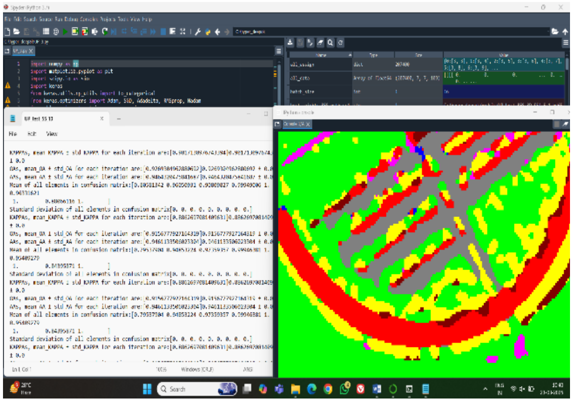
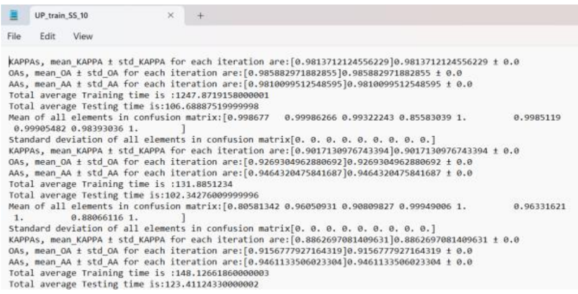


Fig. 3: Recorded Data set of testing "University of Pavia"



UP_train_SS_10

File Edit View

```
KAPPAs, mean_KAPPA ± std_KAPPA for each iteration are:[0.9813712124556229]0.9813712124556229 ± 0.0
OAs, mean_OA ± std_OA for each iteration are:[0.985882971882855]0.985882971882855 ± 0.0
AAs, mean_AA ± std_AA for each iteration are:[0.9810099512548595]0.9810099512548595 ± 0.0
Total average Training time is :1247.8719158000001
Total average Testing time is:106.68887519999998
Mean of all elements in confusion matrix:[0.998677   0.99986266 0.99322243 0.85583039 1.         0.9985119
 0.99905482 0.98393036 1.        ]
Standard deviation of all elements in confusion matrix[0. 0. 0. 0. 0. 0. 0. 0. 0.]
KAPPAs, mean_KAPPA ± std_KAPPA for each iteration are:[0.9017130976743394]0.9017130976743394 ± 0.0
OAs, mean_OA ± std_OA for each iteration are:[0.9269304962880692]0.9269304962880692 ± 0.0
AAs, mean_AA ± std_AA for each iteration are:[0.9464320475841687]0.9464320475841687 ± 0.0
Total average Training time is :131.8851234
Total average Testing time is:102.34276009999996
Mean of all elements in confusion matrix:[0.80581342 0.96050931 0.90809827 0.99949006 1.         0.96331621
 1.         0.88066116 1.        ]
Standard deviation of all elements in confusion matrix[0. 0. 0. 0. 0. 0. 0. 0. 0.]
KAPPAs, mean_KAPPA ± std_KAPPA for each iteration are:[0.8862697081409631]0.8862697081409631 ± 0.0
OAs, mean_OA ± std_OA for each iteration are:[0.915677927164319]0.915677927164319 ± 0.0
AAs, mean_AA ± std_AA for each iteration are:[0.9461133506023304]0.9461133506023304 ± 0.0
Total average Training time is :148.12661860000003
Total average Testing time is:123.41124330000002
```

Fig. 4: Evaluation parameters of University of Pavia

In terms of training and testing time the iterations and the number of pooling layers is varied so as to calculate the average of training and testing time as highlighted in table 3 below:

Table 3: Average training and testing time of dataset university of Pavia

Iteration No.	Pooling Layer	Average Training Time (seconds)	Average Testing Time (seconds)
1	1	1247.87	106.68
2	2	131.89	102.34
3	2	148.13	123.41
4	4	140.01	112.88
5	4	144.07	118.14
6	8	142.04	115.51

7	8	143.06	116.83
8	16	142.55	116.17
9	16	142.80	116.50
10	16	142.67	116.33
11	16	142.74	116.41
12	16	142.71	116.37
13	16	142.72	116.39
14	16	142.71	116.38
15	16	142.72	116.39
16	16	142.72	116.39

In terms of pooling layers of convolution, the algorithm reaches a definitive conclusion that utilizing 16 layers of pooling provided a average training time of 142 ticks and testing time of 116 ticks. Hence RESNET 8 or RESNET 16 can be depended on computational requirements and its feasibilities in terms of personal computer. To increase the accuracy levels, one should increase the number of layers of pooling for classification, but it comes at the cost of testing time which cannot be reduced after a certain threshold is reached. This is in terms of memory management and hardware resource management. To overcome these people have migrated to server-based solutions like that of Google Collab but out research and comparison is not feasible at this developing stage of node computation.
The main focus of this research is to provide noise reduction and improved classification solutions with the help of CGLM based RESNET architecture in terms of reported citations. This cannot be possible unless one can study the impact of noise on original image and how its removal with the help of CGLM has helped in better classification. For this in our research we highlight the two datasets under test in terms of noise and classification as shown in figure 5 below. Table 4 highlights the improvement of our algorithm in comparison to the reported algorithms [1],[17].

Table 4: Accuracy and Loss Comparison

Name of the Dataset	Reference algorithm [44]		Resnet 16 [1]		CGLM RESNET 16	
	Accuracy (%)	Loss (%)	Accuracy (%)	Loss (%)	Accuracy (%)	Loss (%)
Indian Pines	99.02	3.55	99.86	3.3	99.88	2.8
Pavia University	99.94	0.32	99.56	0.74	99.32	0.23

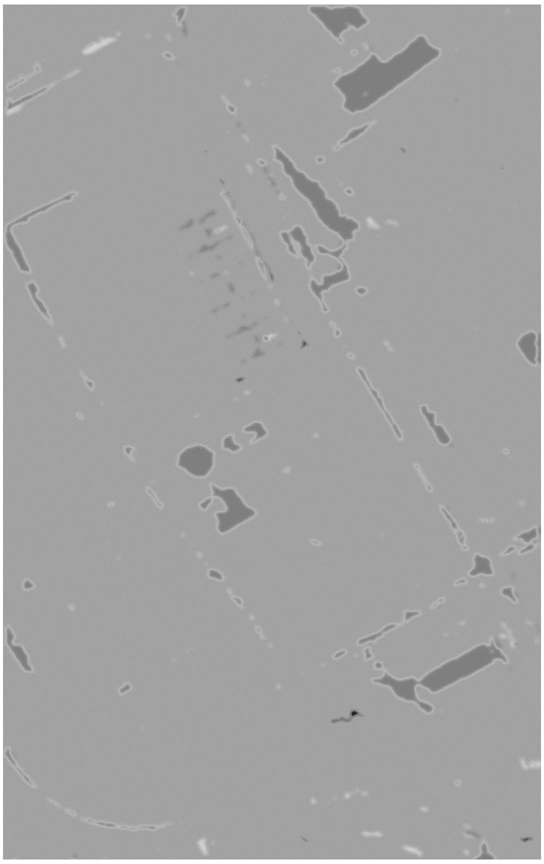


Fig 5.a: University of Pavia Noisy Outlined Image

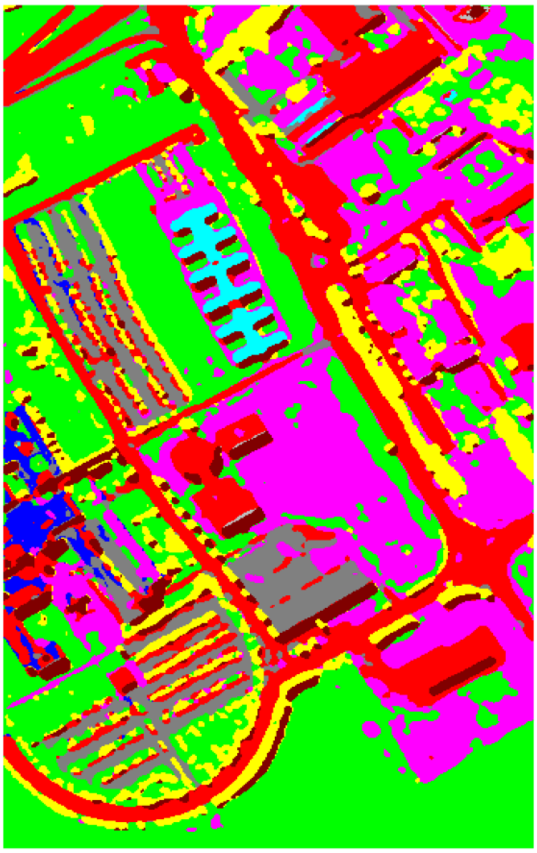


Fig 5.b: University of Pavia Classified Map Image

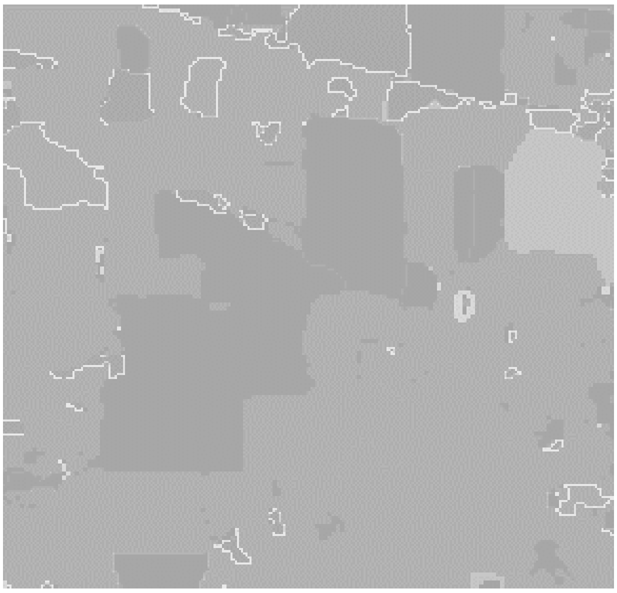


Fig 5.c: Indian Pines Noisy Outlined Image

Fig 5.d: Indian Pines Classified Map Image

5. Conclusion

Based on the above table CGLM tweaked ResNet-16 demonstrates superior performance particularly in terms of accuracy and loss reduction. For the Indian Pines dataset, CGLM ResNet-16 achieves the highest accuracy (99.88%) and the lowest loss (2.8%), outperforming all. CGLM ResNet-16 consistently improves model performance by reducing loss across both datasets, while maintaining competitive accuracy. Statistically the loss parameter is brought down to 2.8 from 3.3 which is 16 % improvement in terms of losses for Indiana pines dataset and 64% for university of Pavia which is better than RESNET. On comparison with reference algorithm the same is 21% for Indiana pines dataset and 29% for university of Pavia. This suggests that CGLM ResNet-16 is more efficient and stable compared to the standard ResNet-16 model. For University of Pavia as well we have achieved enhancing accuracy while maintaining low loss. Applying advanced data augmentation techniques such as GAN-based augmentation or synthetic data generation may help improve model generalization and boost accuracy further.

References

[1] Babrekar V J, and Deshmukh S M (2025) Spatial resolution and noise impact on multi band hyperspectral image processing. *Panamerican Mathematical Journal* 35(2s). https://doi.org/10.52783/pmj.v35.i2s.2571.

[2] Wang Z, and Bovik A C (2009) A universal image quality index. *IEEE Signal Processing Letters*.

[3] Peleg S, and Tishby N (1983) Noise reduction in images using order statistics. *IEEE Transactions on Acoustics, Speech, and Signal Processing*.

[4] Wiener N (1949) *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. MIT Press.

[5] Tomasi C, and Manduchi R (1998) Bilateral filtering for gray and color images. In *IEEE International Conference on Computer Vision (ICCV)*.

[6] Buades A, Coll B, and Morel J M (2005) A non-local algorithm for image denoising. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[7] Donoho D L, and Johnstone I M (1994) Ideal spatial adaptation by wavelet shrinkage. *Biometrika*.

[8] Rudin L I, Osher S, and Fatemi E (1992) Nonlinear total variation-based noise removal algorithms. *Physica D*.

[9] Zhang K, Zuo W, and Zhang L (2017) Learning deep CNN denoiser prior for image restoration. *IEEE Transactions on Image Processing*.

[10] Turk M, and Pentland A (1991) Eigenfaces for recognition. *Journal of Cognitive Neuroscience*.

[11] Perona P, and Malik J (1990) Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[12] Kalman R E (1960) A new approach to linear filtering and prediction problems. *Transactions of the ASME*.

[13] Geman S, and Geman D (1984) Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[14] Kang B, and Kim K (2017) Spatiotemporal image denoising using a deep recurrent convolutional network. *IEEE Transactions on Image Processing*.

[15] Haralick R M, Shanmugam K, and Dinstein I (1973) Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*.

[16] Zhang L, and Shen H (2016) Deep learning for image classification and texture feature extraction. *IEEE Transactions on Image Processing*.

[17] Vaddi R, and Manoharan P (2020) Hyperspectral image classification using CNN with spectral and spatial features integration. *Infrared Physics & Technology* 107: 103296.