Comparative Analysis of Machine Learning Algorithms for Land Cover Classification Using Hyperspectral Images

Mary Nathalie Dela Cruz
Elsa Joy Horiondo

Date of Submission: January 19, 2024

1. ABSTRACT

Land cover classification maps can be derived from hyperspectral images collected with remote sensing techniques via machine learning algorithms. In this study, we aim to determine the most effective machine learning algorithm among Support Vector Machines, K-Nearest Neighbors, and Random Forest for land cover classification of Indian Pines and Salinas datasets. The proposed method uses Principal Component Analysis, Hyperparameter Tuning, and Supervised Classification Algorithms for mapping. The results of the comparative analysis of the three classification methods revealed patterns in training accuracy and error distribution. It was seen that Support Vector Classifiers had an advantage in overall precision and accuracy compared to K-Nearest Neighbors and Random Forest Classifiers. The density and distribution of errors in the plots also suggested that the Support Vector Classifier was more effective for both datasets, possibly due to its proficiency in managing linear separability in higher-dimensional spaces, while the Random Forest Classifier provided the balance between capturing the intricacies of the data and maintaining a level of generalization that avoids over-fitting.

2. INTRODUCTION

Hyperspectral imaging merges spatial imaging and precise chemical spectroscopy to analyze objects digitally. This method is widely used in various sectors, including agriculture, surveillance, mineralogy, biotechnology, and medicine [7]. It's particularly prevalent in geological mapping, which uses remote sensing to generate high-resolution spectral data. Furthermore, this technology excels at identifying and differentiating specific object characteristics, providing markers for distinct chemical and molecular traits inside an image. Such detailed information improves the detection and classification of diseases, anomalies, and essential clues/patterns for researchers, potentially enhancing accuracy.

However, processing data from hyperspectral sensors poses a significant challenge due to its existence in a complex, high-dimensional space. Studies suggest this space is largely unoccupied, with crucial data structures confined to a smaller subset [8]. Additionally, the spectral bands, which are often interconnected, may carry redundant information. The presence of noisy and irrelevant bands further complicates the computational demands of analyzing hyperspectral images [3].

This study aims to use different classification methods to determine the most appropriate machine-learning algorithm for land cover remote sensing data of Indian Pines and Salinas. This research investigates the most effective classification algorithm among Support Vector Machines, K-Nearest Neighbors, and Random Forest for land cover classification utilizing hyperspectral image data derived from remote sensing techniques, focusing on Indian Pines and Salinas datasets. This study further aims to accomplish specific objectives, including optimizing hyperparameters for each of the aforementioned algorithms and assessing their performance through evaluation metrics.

3. METHODOLOGY

The proposed method uses Principal Component Analysis, Hyperparameter Tuning, and Supervised Classification Algorithms for analyzing hyperspectral images of Indian Pines and Salinas.

3.1 Data Description

To conduct the study, two hyperspectral images were downloaded from the UPV/EHU website: the Indian Pines hyperspectral image and the Salinas hyperspectral image. [1]

3.1.1 Indian Pines

The Indian Pines hyperspectral image was collected by the Airborne Visible Infrared Imaging Spectrometer (AVIRIS) sensor over the Indian Pines test site in North-Western Indiana, USA. After removing the noise and water absorption bands, it has 200 bands with a spatial resolution of 20-meter pixels and a spatial extent of 145 pixels by 145 pixels per band. Figures 1(a), 1(b), and 1(c) show a sample band of the Indian Pines hyperspectral image, its ground truth classification map, and the overlay of the ground truth data and the sample band. The available ground truth data is divided into 16 classes representing agriculture, forest, or natural perennial vegetation in the area. The known land cover classes are shown in Table 1 along with the number of samples per class. [1]

3.1.2 Salinas

The Salinas hyperspectral image was collected by the 224-band AVIRIS sensor over the Salinas Valley, California. Same as the Indian Pines hyperspectral image, its noise and water absorption bands were removed and it now has 204 bands with a spatial resolution of 3.7-meter pixels and a spatial extent of 512 pixels by 217 pixels per band. A sample band of the Salinas hyperspectral image, its ground truth classification map, and the overlay of the ground truth data and

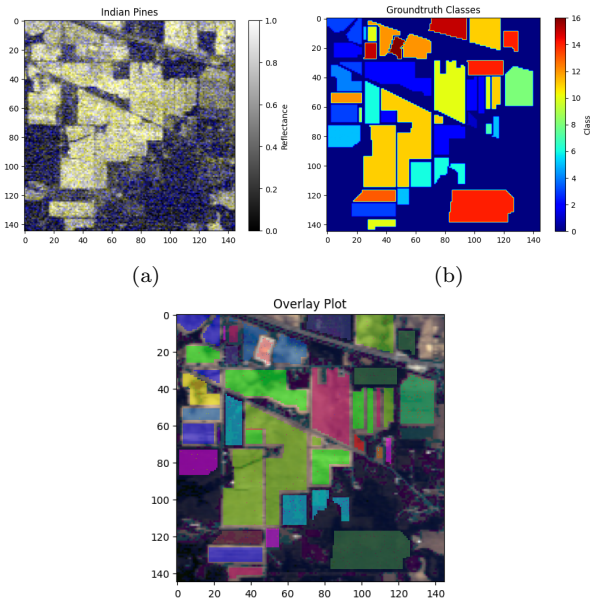


Figure 1: (a) Visualization of the Indian Pines hyperspectral image. (b) Ground truth classification map for the Indian Pines hyperspectral image where each color represents a different class or land cover. (c) Overlay of the hyperspectral image and the ground truth classification for visual comparison of the spectral data and known land cover classes.

Class	Name	Samples
1	Alfalfa	46
2	Corn Notill	1428
3	Corn Mintill	830
4	Corn	237
5	Grass Pasture	483
6	Grass Trees	730
7	Grass Pasture Mowed	28
8	Hay Windrowed	478
9	Oats	20
10	Soybean Notill	972
11	Soybean Mintill	2455
12	Soybean Clean	593
13	Wheat	205
14	Woods	1265
15	Building Grass Trees Drives	386
16	Stone Steel Towers	93

Table 1: Known land cover classes and the corresponding number of samples of Indian Pines Hyperspectral Image.

the sample band are shown in Figures 2(a), 2(b), and 2(c). The available ground truth data is divided into 16 classes representing agriculture, soils, and the vineyard field in the area. The known land cover classes are shown in Table 2 along with the number of samples per class. [1]

3.2 Proposed Method

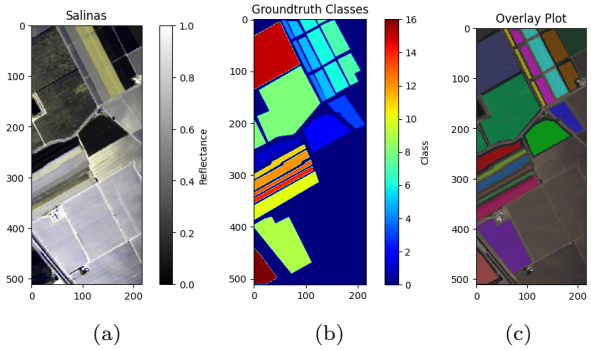


Figure 2: (a) Visualization of the Salinas hyperspectral image. (b) Ground truth classification map for the Salinas hyperspectral image where each color represents a different class or land cover. (c) Overlay of the hyperspectral image and the ground truth classification for visual comparison of the spectral data and known land cover classes.

Class	Name	Samples
1	Broccoli Green Weeds 1	2009
2	Broccoli Green Weeds 2	3726
3	Fallow	1976
4	Fallow Rough Plough	1394
5	Fallow Smooth	2678
6	Stubble	3959
7	Celery	3579
8	Grapes Untrained	11271
9	Soil Vineyard Develop	6203
10	Corn Senesced Green Weeds	3278
11	Lettuce Romaine 4 wk	1068
12	Lettuce Romaine 5 wk	1927
13	Lettuce Romaine 6 wk	916
14	Lettuce Romaine 7 wk	1070
15	Vineyard Untrained	7268
16	Vineyard Vertical Trellis	1807

Table 2: Known land cover classes and the corresponding number of samples of Salinas Hyperspectral Image.

An overview of the proposed methodology is shown in Figure 3. The framework is divided into two parts: Preliminary Processing, and Modeling.

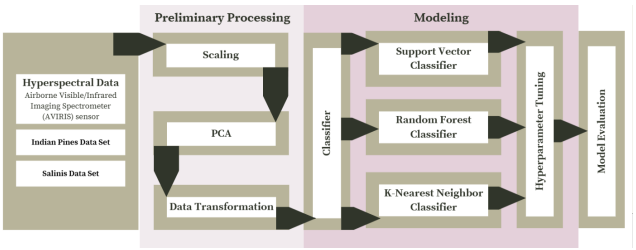


Figure 3: Framework for the proposed methodology. The process is divided into two parts: Preliminary Processing, and Modeling.

Preliminary processing is often necessary when working with hyperspectral images. First, the pixel intensities of the hyperspectral image were scaled to maintain zero mean and

unit variance. The key component in this framework is to perform principal component analysis on the scaled hyperspectral image. Here, principal component analysis reduces the high dimensionality of its spectral features. Then the pre-processed hyperspectral image was converted from a 3-dimensional image cube to a 2-dimensional structured format for easier analysis.

The goal of modeling is to build classification maps of the location in the hyperspectral image using a classifier function. The classifier function supports 3 types of Supervised Machine Learning algorithms: Support Vector Classification, K-Nearest Neighbors Classification, and Random Forrest Classification. The hyperparameters of each model were tuned using GridSearchCV, where the selected hyperparameters managed to produce the model with the highest accuracy.

3.2.1 *Preliminary Processing*

Before modeling, it is common practice to normalize data to follow typical assumptions that make classifiers work better, such as having zero mean and unit variance. This also ensures that equal weights are given to spectral features regardless of their inherent large value. The standard strategy for normalization is often statistical in approach and so, in this study, the first-order and the second-order moments of the hyperspectral image were normalized to obtain a zero mean and unit variance. This is done globally on the whole hyperspectral image with the equation,

$$I = \frac{I - \mu_I}{\sigma_I} \tag{1}$$

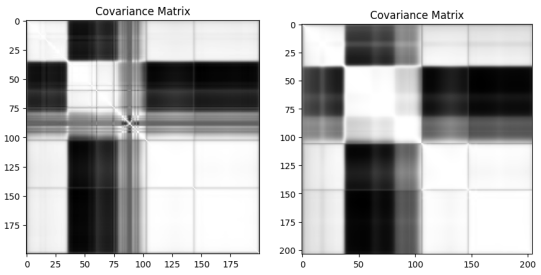
where I represents the hypespectral image, μ_I is the mean, and σ_I is the standard deviation of the hyperspectral image. [2, 4]

Since the spectral features are scaled, principal component analysis can now be used to select bands and drop uninformative bands. Principal component analysis is one of the most commonly used dimensionality reduction techniques. It is often used on 2-dimensional matrices. To apply the principal component analysis on a 3-dimensional hyperspectral image, the covariance matrix must be computed from its features. The covariance between two feature vectors X and Y representing spectral bands x and y is calculated with the equation,

$$Cov(X, Y) = \sum \frac{(X_i - \bar{X})(Y_i - \bar{Y})}{n - 1} \tag{2}$$

where $\bar{X}$ and $\bar{Y}$ are the respective mean of X and Y, and n is the number of pixels. Figures 4(a) and 4(b) show the covariance matrix of Indian Pines and Salinas, respectively. In these matrix displays, covariance is more positive for lighter shades, more negative for darker shades, and near zero for gray shades. [2, 3]

The covariance matrix is decomposed into its eigenvalues and eigenvectors with the equation,



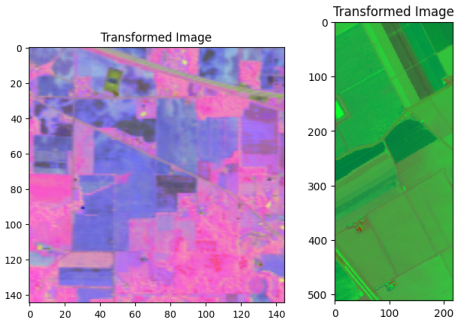
(a) (b)

Figure 4: Visualization of the covariance matrix of the principal component derived from (a) Indian Pines Hyperspectral Image and (b) Salinas Hyperspectral Image.

$$kv = \lambda v \tag{3}$$

where k represents the eigenvector, λ represents the diagonal matrix, and v represents the covariance matrix. Here, the direction of the eigenvector has a variance equal to the corresponding eigenvalue. The calculated eigenvalues are sorted from highest to lowest, where a necessary number of eigenvalues are kept to preserve a certain percentage of the covariance matrix, forming a feature matrix. The dimensionality of the original hyperspectral image was reduced by projecting it onto the feature matrix through the multiplication of the transposition of the feature matrix and the transposition of the original hyperspectral image. [2, 3]

In this study, a minimum of 99.9% of the total image variance was retained. Figure 5(a) and 5(b) shows the first three principal components of the Indian Pines hyperspectral image and Salinas hyperspectral image, respectively, after the matrix projection.



(a) (b)

Figure 5: Visualization of the first three principal components of the transformed (a) Indian Pines Hyperspectral Image and transformed (b) Salinas Hyperspectral Image.

With the principal component analysis, the number of spectral bands of the Indian Pines hyperspectral image was reduced from 200 bands to 108 bands while the number of spectral bands of Salinas hyperspectral image was reduced from 204 bands to 19 bands. After pre-processing, the hyperspectral image was transformed from a 3D image cube to a 2D structured format for simpler analysis.

3.2.2 *Modeling Framework*

For the classification of different land covers of Indian Pines and Salinas hyperspectral images, three supervised machine learning algorithms were selected: Support Vector Classifier, Random Forest Classifier and K-Nearest Neighbors Classifier.

In simple terms, Support Vector Machines (SVM) act as a classifier for an instance by finding the hyperplane that best separates different classes in a dataset. K-Nearest Neighbors (KNN), on the other hand, determines the K closest instances or the neighbors of the new instance. It then assigns the class of an instance based on the most common class of its neighbors. Meanwhile, Random Forest Classifier (RFC) determines the class of an instance by constructing multiple decision trees during training.[5]

In the classification algorithm, a grid of hyperparameters was initialized for each model. For each point in the grid, the hyperparameters of each model were tuned. Cross-validation is performed for each combination of hyperparameters in the grid, where the model is repeatedly trained on some splits of the training data and evaluated on the remaining splits. After fitting, the optimized parameters of a model are obtained from the grid search based on the most accurate model during the cross-validation. [6] Table 3 shows the optimized parameters obtained for each classification algorithm for Indian Pines hyperspectral image. Table 4 shows the optimized parameters obtained for each classification algorithm for the Salinas hyperspectral image.

Model	Hyperparameter	Value	Accuracy
SVC	regularization	1000	83.85%
	kernel	rbf	
	gamma	scale	
RFC	number of trees	200	71.85%
KNN	metric	manhattan	73.41%
	number of neighbors	9	
	weights	distance	

Table 3: Optimized parameters for Indian Pines hyperspectral image of three classification algorithms: Support Vector Classifier (SVC), Random Forest Classifier (RFC), and K-Nearest Neighbors (KNN) after applying a hyperparameter tuning technique.

Model	Hyperparameter	Value	Accuracy
SVC	regularization	1000	91.98%
	kernel	rbf	
	gamma	scale	
RFC	number of trees	230	91.95%
KNN	metric	manhattan	90.75%
	number of neighbors	11	
	weights	distance	

Table 4: Optimized parameters for Salinas hyperspectral image of three classification algorithms: Support Vector Classifier (SVC), Random Forest Classifier (RFC), and K-Nearest Neighbors (KNN) after applying a hyperparameter tuning technique.

4. ANALYSIS AND DISCUSSION OF RESULTS

4.1 Classifier Comparison Results in Indian Pines

The comparative analysis of the three classification methods – Support Vector Classifier (SVC), k-Nearest Neighbors (kNN), and Random Forest Classifier (RFC) – revealed distinct patterns in training accuracy and error distribution.

For the Indian Pines data, as seen in Figure as seen in Figure 6, the SVM classification map exhibited intricate decision boundaries with numerous micro-regions, indicating a highly fitted model. The associated training errors are minimal, signaling a strong adherence to the training data, which may suggest over-fitting tendencies. On the other hand, KNN presented fragmented boundaries, portraying the default approach of localized averaging. The training error showed a greater dispersion of misclassified sampling points than SVC, which implied less precision. However, it could potentially draw better generalizability compared to SVM. RFC gave block-like decision boundaries reflecting the ensemble approach of several decision trees. Its error plot was more scattered than SVM but was more precise than KNN. Thus displaying a balanced model complexity.

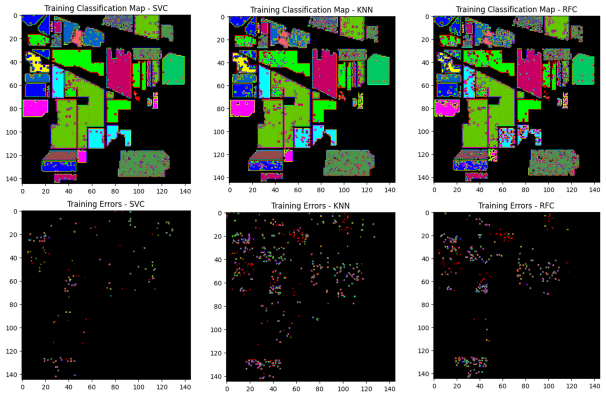


Figure 6: Classification Maps and Errors of SVC, kNN, RFC for Indian Pines.

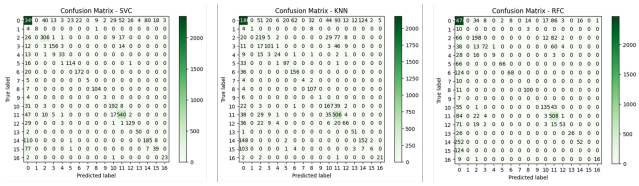


Figure 7: Confusion Matrix of SVC, kNN, RFC for Indian Pines.

For KNN and RFC models, only class 11 excels notably with an impressive 506 and 508 correct predictions, while classes 2 and 11 correctly predicted 306 and 540 (the highest across all algorithms for any class) instances for the SVC (Figure 7). It can also be seen that the SVC seemed to have a more homogeneous average precision (Figure 8) and accuracy (83.85%) compared to KNN (73.41%) and RFC (71.85%) (Figure 9).

Figure 10 compared the classification performance of three machine learning algorithms against a true map using a

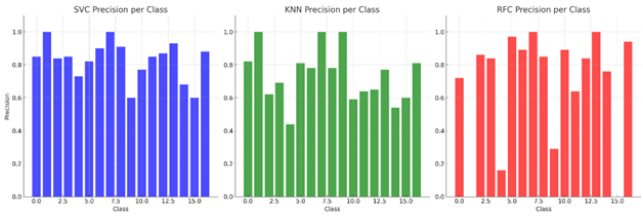


Figure 8: Precision per Class of SVC, kNN, RFC for Indian Pines.

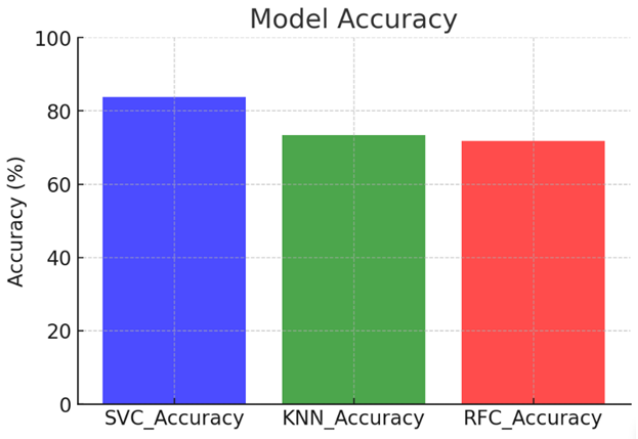


Figure 9: Model Accuracy of SVC, kNN, RFC for Indian Pines.

color-coded classification system ranging from 0 to 16 to represent various classes or categories. In the first pair, the SVC's output displayed a marked deviation from the true map, with a tendency towards over-generalization, which led to inaccurately represented class boundaries and an over-representation of certain classes. Conversely, the KNN classification map indicated a potential over-fitting issue, evidenced by a noisy pattern with many small, incorrectly classified regions that suggest the algorithm may be picking up on random noise as if it were significant. The RFC's output provided a more balanced classification, though not without errors. It offered a compromise between the smooth generalization of the SVC and the noisy precision of the KNN, resulting in a map that more closely approximates the true reference but still contained regions of misclassification.

4.2 Classifier Comparison Results in Salinas Pines

A second data source, the Salinas dataset, was utilized to further evaluate the effectiveness of the three classifiers. After processing, the hypercubes from Salinas were found to be cleaner and more normalized than those from the Indian Pines dataset, offering improved insights into the discriminative abilities of the classifiers. The results in Figure 11 compare the performance of the SVC, KNN, and RFC. In summary, the density and distribution of errors in the plots suggest that SVC was more effective for this dataset, possibly due to its proficiency in managing linear separability in higher-dimensional spaces. The performance of KNN may have been affected by over-fitting or an inappropriate choice

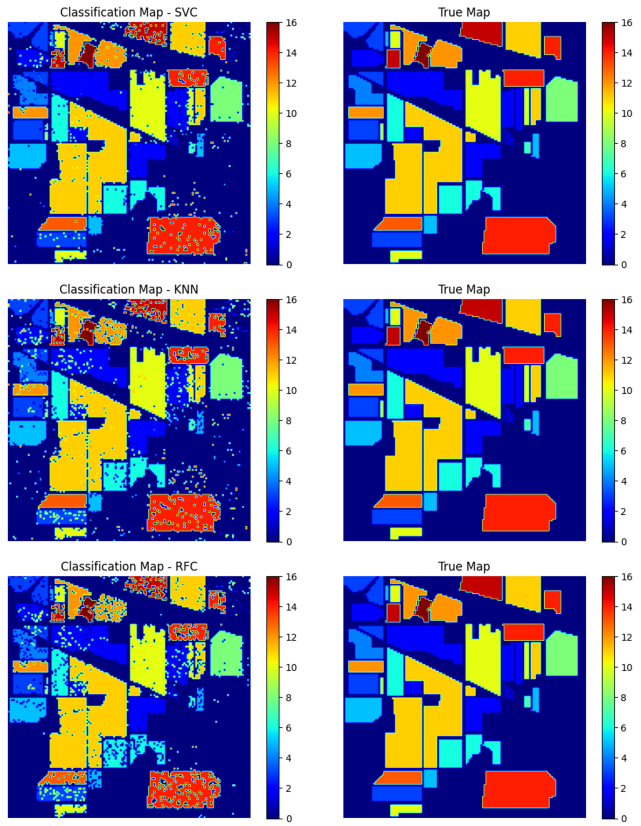


Figure 10: Classification Map vs True Map of SVC, kNN, RFC for Indian Pines.

of 'K'. Meanwhile, the Random Forest algorithm demonstrated a balance of robustness, although it did not surpass SVC's performance in this scenario.

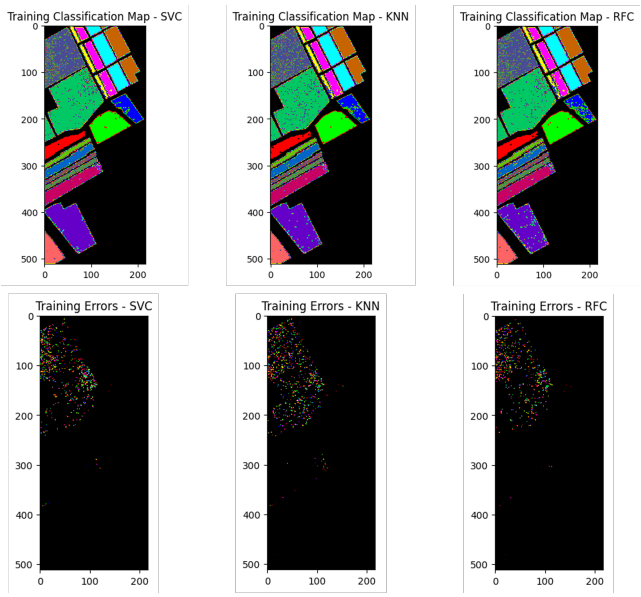


Figure 11: Classification Maps and Errors of SVC, kNN, RFC for Salinas.

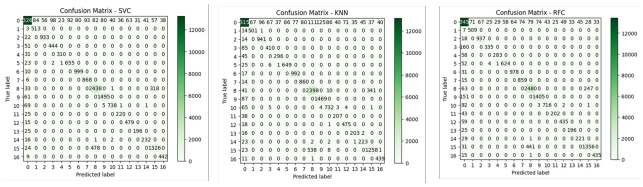


Figure 12: Confusion Matrix of SVC, kNN, RFC for Salinas.

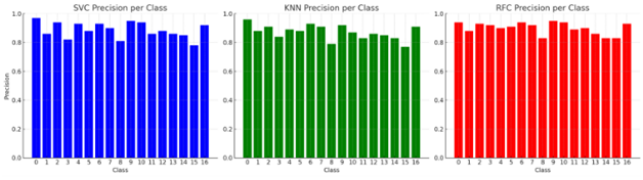


Figure 13: Precision per Class of SVC, kNN, RFC for Salinas.

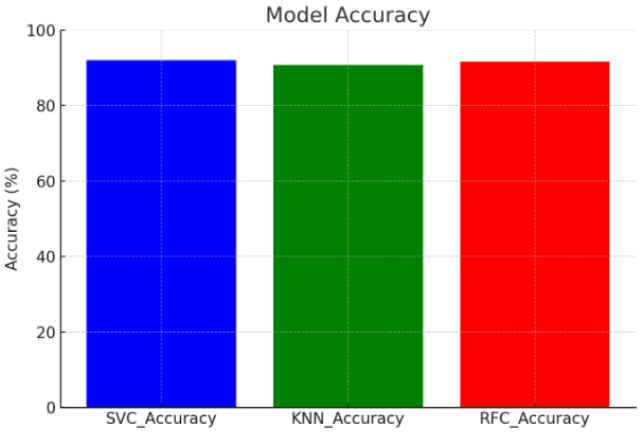


Figure 14: Model Accuracy of SVC, kNN, RFC for Salinas.

When comparing the data from Indian Pines with that of Salinas, the confusion matrices from Salinas indicated a five-fold increase, as shown in Figure 12. This result is corroborated by the high overall precision observed across all classes. In this analysis, the three classifiers used demonstrated average precision ranging between 77% and 97%, and accuracies exceeding 90% (SVC at 91.98%, KNN at 90.75%, RFC at 91.55%).

Like in Indian Pines, the classification performance of three algorithms was also compared for the Salinas dataset. The SVC algorithm's map was characterized by broad swathes of color, suggesting a high degree of generalization. While some correspondence with the true map was evident, the SVC's output lacked detail, and the boundaries between classes were not well-delineated, hinting at a possible high bias in the model, which might lead to under-fitting. The KNN algorithm provided a contrasting output with a mixture of large blocks of color interspersed with speckled areas, pointing towards an attempt to capture both general trends and detailed variations. However, this approach resulted in a map that neither perfectly mirrors the true map's complexity nor avoids noise, indicating a potential compromise between bias and variance. Lastly, the RFC algorithm produced a map that closely mirrors the true map, with better-defined boundaries and a higher level of detail, suggesting that it managed to achieve a balance between capturing the intricacies of the data and maintaining a level of generalization that avoids over-fitting.

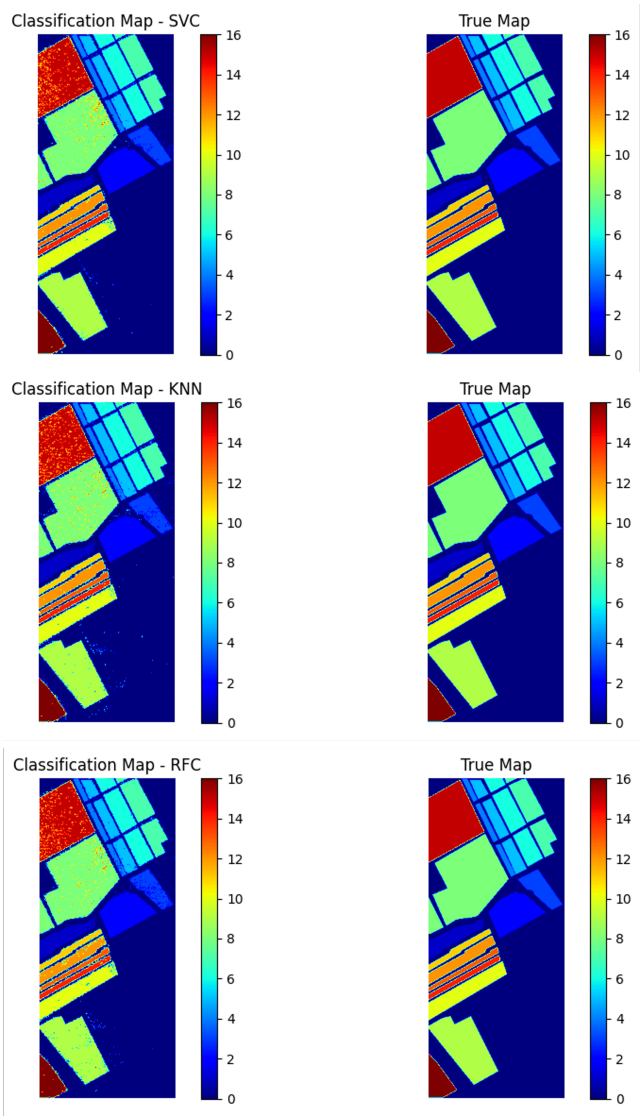


Figure 15: Classification Map vs True Map of SVC, kNN, RFC for Indian Pines.

5. CONCLUSION

In hyperspectral imaging applications such as land cover classification from satellite imagery, the choice of an algorithm would significantly impact the accuracy and usability of the data. In both Indian Pines (imbalanced) and Salinas (balanced) hyperspectral data, the support vector classifier exemplified promising results in terms of its lower training errors, better overall precision among classes, and higher accuracies. Hence, the SVC's tendency to generalize might benefit broader classifications where the primary interest lies in distinguishing between large, distinct areas. The KNN could be considered in scenarios where detail is essential,

but one must be cautious of the noise it might introduce into the classification. The choice of algorithm would hinge on the specific application and importance of accurately identifying certain classes over others, but if a map that has a balance between over-fitting and under-fitting were preferred, the RFC would likely be the safer model for tasks requiring precise demarcation of classes and serve as a verified benchmark.

6. REFERENCES

[1] EHU, "Hyperspectral Remote Sensing Scenes," available at: https://ehu.eus/ccwintco/index.php?title=Hyperspectral Remote_Sensing_Scenes.

[2] M.R. Haque and S.Z. Mishu, "Spectral-Spatial Feature Extraction Using PCA and Multi-Scale Deep Convolutional Neural Network for Hyperspectral Image Classification" in *2019 22nd International Conference on Computer and Information Technology (ICCIT)*, 2019, pp. 1-6.

[3] Spectral Python, "Spectral Algorithms," available at: https://www.spectralpython.net/algorithms.htmlk-means-clustering.

[4] N. Audebert, B. Le Saux and S. Lefevre, "Deep Learning for Classification of Hyperspectral Data: A Comparative Review," in *IEEE Geoscience and Remote Sensing Magazine, vol. 7, no. 2*, 2019, pp. 159-173.

[5] Bonaccorso, Giuseppe. Machine learning algorithms. Packt Publishing Ltd, 2017.

[6] Sci kit Learn, "Cross-validation: evaluating estimator performance," available at: https://scikit-learn.org/stable/modules/cross$_v$*alidation.html*

[7] A. Elmaizi, E. Sarhrouni, A. Hammouch, and C. Nacir, "A new band selection approach based on information theory and support vector machine for hyperspectral images reduction and classification," in *2017 International Symposium on Networks, Computers and Communications (ISNCC)*, 2017, pp. 1-6.

[8] A. Moghimi, C. Yang, and P. M. Marchetto, "Ensemble feature selection for plant phenotyping: A journey from hyperspectral to multispectral imaging," *IEEE Access*, vol. 6, pp. 1-1, 2024. [Online]. Available: https://doi.org/10.1109/ACCESS.2024.2872801.

[9] A. Plaza, P. Martinez, J. Plaza, and R. Perez, "Dimensionality reduction and classification of hyperspectral image data using sequences of extended morphological transformations," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 466-479, Mar. 2005. [Online]. Available: https://doi.org/10.1109/TGRS.2004.841417.