Comparative Study of Machine Learning Techniques for Land Cover Classification and Mapping

Amey Parle, Sanika Marathe, Swamini Sontakke

ABSTRACT:

Land cover classification and mapping are critical for environmental monitoring, urban planning, and sustainable development. Precise and scalable predictions of land cover enable decision-makers to have a data-driven understanding of how land is used and how it may evolve, leading to better policies, strategic resource allocation, and sustainable development. This study compares the performance of several machine learning techniques, including Support Vector Machines (SVM), Deep Learning (CNN) and Transfer Learning (VGG-16) for accurately classifying land cover in geospatial images. Using a benchmark dataset derived from satellite imagery, the algorithms were evaluated based on classification accuracy, computational efficiency, and scalability. The results demonstrate that deep learning methods outperform traditional techniques in accuracy, albeit with higher computational demands. This research provides insights into selecting appropriate machine learning models for land cover analysis in diverse scenarios.

KEYWORDS: Data Mining, Machine Learning, Transfer Learning, Land Cover Classification

INTRODUCTION:

The rapid expansion of urban areas, agricultural activities, and environmental changes has made the need for accurate and efficient land cover classification more critical than ever. With the increasing availability of satellite imagery, accurately classifying land cover types has become achievable. Traditional machine learning, advanced deep learning and Transfer Learning methods have shown promising results in image classification tasks.

However, the choice of ML techniques impacts classification accuracy, computational cost, and practical implementation.

The project proposes conducting a comparative study of various image classification algorithms. This study has significant potential applications in fields such as urban planning, resource management, environmental monitoring, and geospatial analysis, enabling informed decision-making for sustainable development and effective resource allocation.

DATASET:

In this project we have used the EuroSAT dataset which consists of 27000 RGB Images of various types of land covers. These images have been captured by the Sentinel2 satellite and have dimensions of 64x64 pixels with a Ground Sampling Distance of 10m. All these images have been categorized in 10 different classes.

This dataset is split into a training set which constitutes 70% (18900 images) of the entire dataset, a validation set that constitutes 20% and a test set containing 10% images of the entire dataset. We also have 3 csv files for each set containing paths to images belonging to the respective set.

Dataset Link: https://www.kaggle.com/datasets/apollo2506/eurosat-dataset

METHODS:

In this project we have chosen algorithms from each of the 3 different technologies, viz, Machine Learning, Deep Learning and Transfer Learning.

1. **Machine Learning:** From the current advanced Machine Learning algorithms, we choose to study the performance of the Support Vector Machine (SVM).

1.1 Process Followed:

1.1.1 Data Preprocessing:

1.1.1.1 Image resizing and Feature extraction:

A pre-trained VGG16 model is initialized without its fully connected layers, allowing it to function as a feature extractor. The input size is set to 128x128 pixels for RGB images.
Images are resized to 128x128 pixels and converted into numerical arrays. They are then preprocessed to match the input format expected by the VGG16 model.

The preprocessed images are passed through the VGG16 model to extract feature maps from the last convolutional layer. These feature maps are flattened into feature vectors, enabling compatibility with the SVM classifier.

1.1.2 Model training

An SVM classifier with a radial basis function (RBF) kernel is selected for its ability to handle high-dimensional data. The classifier is trained using the extracted feature vectors and corresponding labels.

1.2 Rationale for the Approach

1.2.1 Pre-trained Model Usage: The VGG16 model provides high-level features, reducing the need for manual feature engineering.

1.2.2 Dimensionality Reduction: The feature extraction simplifies complex image data into a manageable format.

1.2.3 SVM Classifier: The SVM effectively classifies high-dimensional feature vectors, ensuring robust model performance.

1.3 Results

1.3.1 Classification report

	precision	recall	f1-score	support
AnnualCrop	0.93	0.88	0.90	300
Forest	0.93	0.98	0.95	300
HerbaceousVegetation	0.89	0.90	0.89	300
Highway	0.79	0.73	0.76	250
Industrial	0.65	0.97	0.78	250
Pasture	0.91	0.93	0.92	200
PermanentCrop	0.91	0.76	0.83	250
Residential	0.97	0.92	0.94	300
River	0.86	0.71	0.78	250
SeaLake	0.99	0.96	0.97	300
accuracy			0.88	2700
macro avg	0.88	0.87	0.87	2700
weighted avg	0.89	0.88	0.88	2700

Figure 1: Classification Report - SVM

1. **High-Performing Classes:**
   - SeaLake (F1-score: 0.97) and Forest (F1-score: 0.95) achieved near-perfect precision and recall. This indicates the model can effectively identify water bodies and forests, reducing the risk of misclassifying these crucial land types.
   - Residential (F1-score: 0.94) and AnnualCrop (F1-score: 0.90) also performed well, reflecting the model's reliability in distinguishing between human settlements and cultivated lands.
2. **Moderate-Performing Classes:**
   - Industrial (F1-score: 0.78) and River (F1-score: 0.78) had lower recall scores, suggesting the model struggled to detect these classes consistently. This may be due to overlapping features with similar-looking categories like Highway or SeaLake.
   - PermanentCrop (F1-score: 0.83) showed a recall drop, meaning some permanent crops were missed. This could lead to underestimation of certain agricultural areas.
3. **Problematic Areas:**
   - Highway (F1-score: 0.76) exhibited limited performance due to visual similarities with Industrial and Residential areas, causing false positives.

- River (Recall: 0.71) had the lowest recall, indicating that many rivers were not detected correctly, likely due to narrow or visually complex river segments.

1.3.2 ROC-AUC Curve

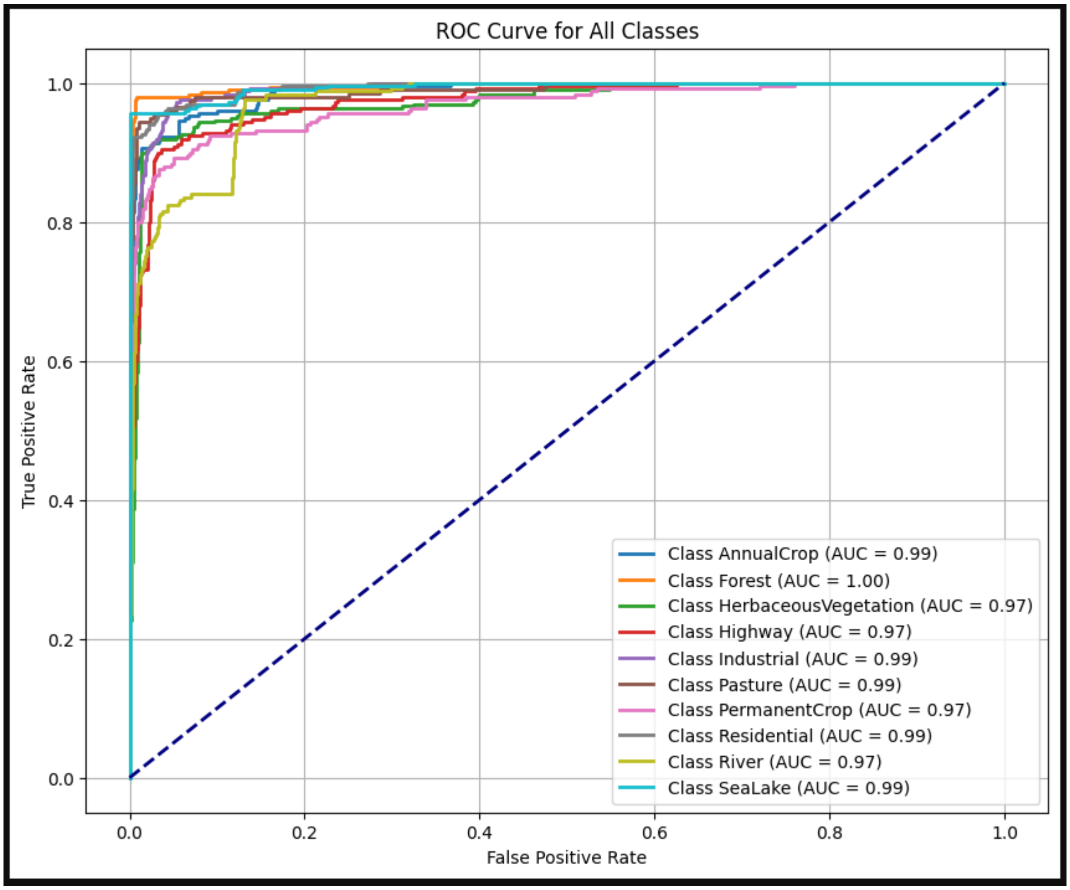

Figure 2: ROC-AUC Curve for all classes- SVM

- The ROC curve illustrates the classifier's strong performance across land-cover classes, with AUC values ranging from 0.97 to 1.00.
- "Forest" achieves a perfect AUC of 1.00, while other classes like "AnnualCrop," "Industrial," and "Residential" score 0.99, showing near-perfect distinction. Classes such as "HerbaceousVegetation," "Highway," "PermanentCrop," and "River" have slightly lower AUCs of 0.97, indicating minor overlap or confusion with similar categories.
- Overall, the model performs excellently, with only slight challenges in distinguishing certain classes, suggesting areas for potential improvement in feature separation for classes like "Highway" and "River."

1.3.3 Confusion Matrix

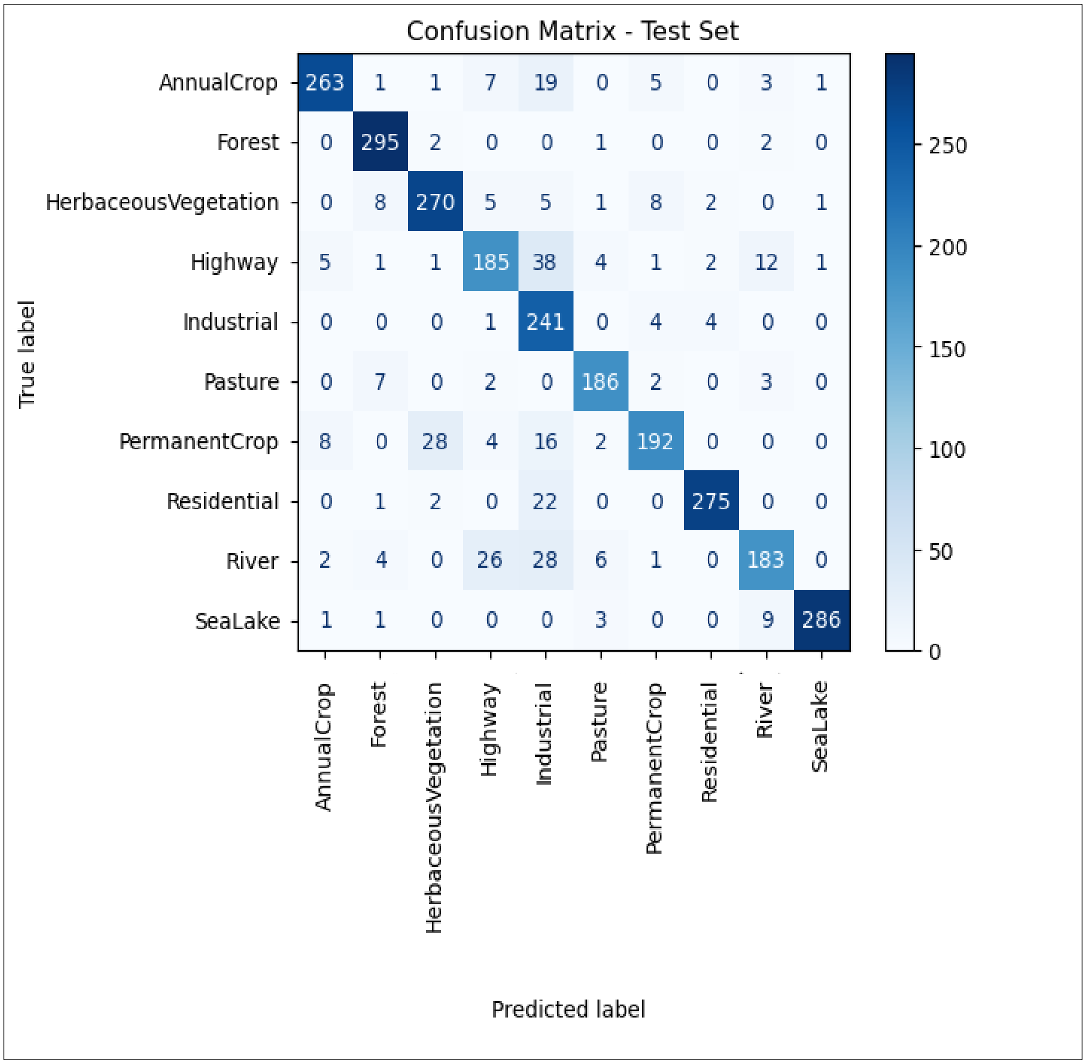

Figure 3: Confusion Matrix - SVM

- The confusion matrix evaluates the SVM Classifier's performance on the test set, with true labels on the y-axis and predicted labels on the x-axis.
- Strong performance is evident for classes like AnnualCrop, Forest, HerbaceousVegetation, Industrial, Residential, and SeaLake, where most predictions align with true labels, as shown by high diagonal values.
- However, misclassifications occur, particularly between similar classes, such as PermanentCrop being confused with HerbaceousVegetation and Pasture, and Highway being misclassified as Residential or other categories.
- The River class also shows significant misclassification with HerbaceousVegetation. While the model demonstrates high accuracy for several classes, improvements are needed in distinguishing closely related categories.

2. **Deep Learning:** From a wide range of Neural Networks, we choose to study the Convolution Neural Networks (CNNs). CNN is well known for working with images and is used in applications related to image classification, segmentation and recognition.

   2.1. Process Followed:

   2.1.1. Data Preprocessing: Before feeding the images to the neural network, images were preprocessed by implementing the following:

   2.1.1.1. Converting images to tensor: By doing this, RGB images were converted to a 3X3 matrix format containing pixel values

   2.1.1.2. Resized the images: Images are resized before feeding to the neural network.

   2.1.1.3. Adjusting Brightness and contrast: Satellite images have different levels of brightness and contrast, hence, for uniformity this modification.

   2.1.1.4. Normalization: Dividing pixel values by 255.

2.1.2. Model Training:

2.1.2.1. Model Architecture: Following diagram represents the architecture of the CNN model. The input shape is (64,64,3).

Model: "functional"

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 64, 64, 3)	0
conv2d (Conv2D)	(None, 64, 64, 32)	896
max_pooling2d (MaxPooling2D)	(None, 32, 32, 32)	0
conv2d_1 (Conv2D)	(None, 32, 32, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 16, 16, 64)	0
conv2d_2 (Conv2D)	(None, 16, 16, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 8, 8, 128)	0
dropout (Dropout)	(None, 8, 8, 128)	0
conv2d_3 (Conv2D)	(None, 8, 8, 128)	147,584
max_pooling2d_3 (MaxPooling2D)	(None, 4, 4, 128)	0
dropout_1 (Dropout)	(None, 4, 4, 128)	0
conv2d_4 (Conv2D)	(None, 4, 4, 64)	73,792
max_pooling2d_4 (MaxPooling2D)	(None, 2, 2, 64)	0
conv2d_5 (Conv2D)	(None, 2, 2, 32)	18,464
max_pooling2d_5 (MaxPooling2D)	(None, 1, 1, 32)	0
global_average_pooling2d (GlobalAveragePooling2D)	(None, 32)	0
dense (Dense)	(None, 64)	2,112
dense_1 (Dense)	(None, 10)	650

Total params: 335,850 (1.28 MB)
Trainable params: 335,850 (1.28 MB)
Non-trainable params: 0 (0.00 B)

Figure 4: Model Architecture/Parameters

- Used MaxPooling for reducing spatial dimensionality of the images.
- Used Dropout layer for preventing overfitting
- Used Global Average Pooling for reducing each feature map to a single number.

2.1.2.2. Hyperparameters:

- Activation Function ReLU used in inner layers ensures that negative values are replaced with zeros, introducing non-linearity and sparsity into the model.
- Activation Function Softmax in the output layers is used since this is a multi-class classification.
- Number of epochs defined:100
- Batch size: 32
- Early Stopping used with patience equal to 5, training stopped at epoch at 83. (monitoring validation loss)
- LR Scheduler: ReduceLROnPlateau used with patience of 5 (monitoring validation loss)
- Used Checkpoints to save best model weights
- Used Adam as the optimizer as it handles sparse gradients.

2.1.2.3. Optimizer & Loss: Calculated sparse_categorical_entropy loss, since the labels are integer encoded.

2.1.2.4. Results:

2.1.2.4.1. Training and Validation Loss

Following Images represent the training loss (Figure 5) and validation loss (Figure 6), for measuring accuracy, over 83 epochs.

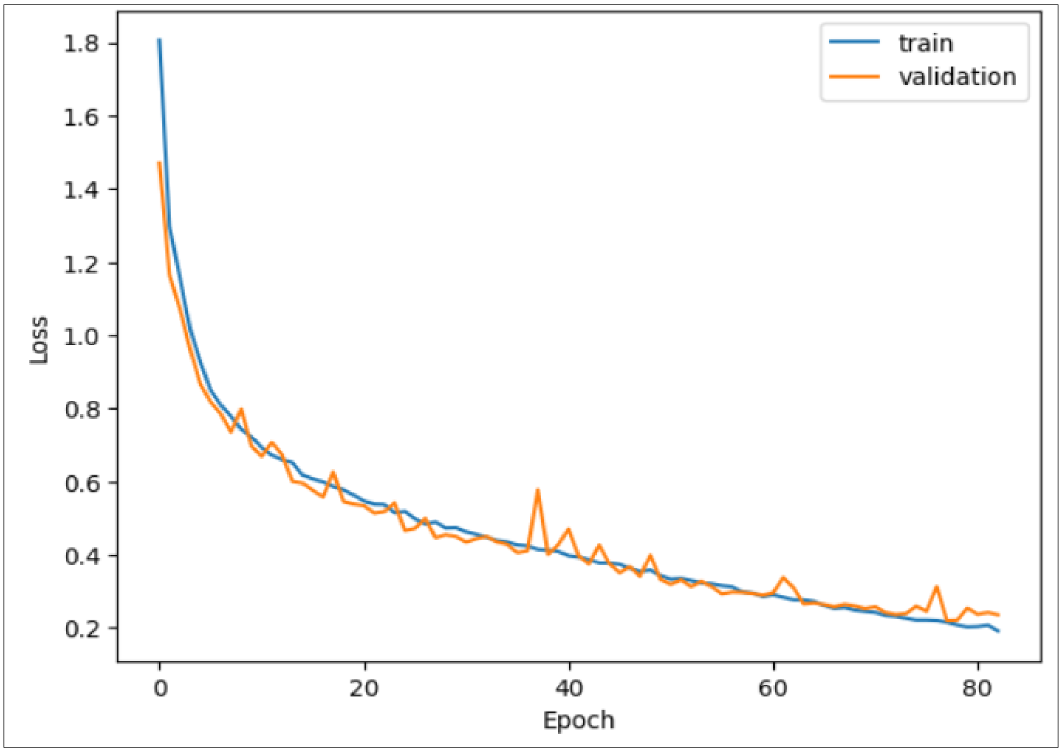

Figure 5: Training/Validation Loss

Figure 5 shows both the training and validation loss decline together; Validation loss is a bit noisy. Both the training and validation loss remain close to each other, indicating there is no significant overfitting. This indicates that the model generalizes well.

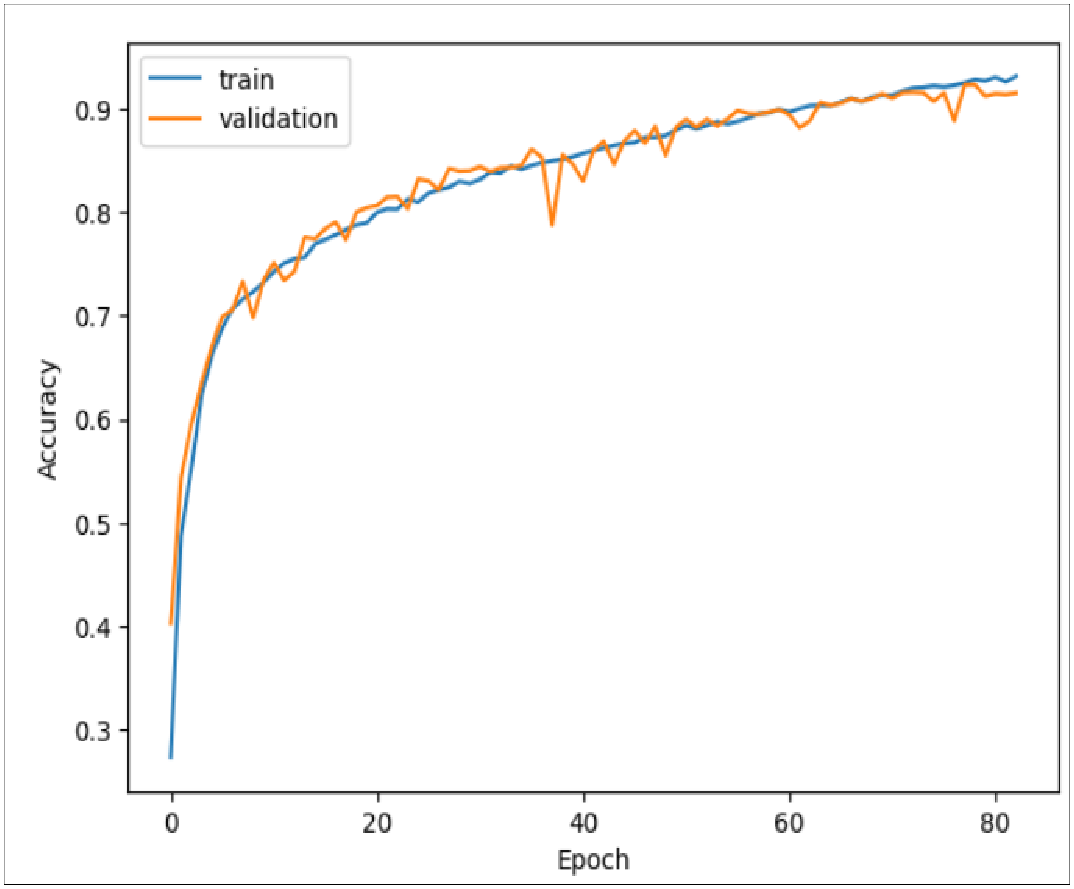

Figure 6: Training/ Validation Accuracy

Figure 6 shows both the training and validation accuracy increase gradually together with the increasing number of epochs. Up to 10 epochs the accuracy increases rapidly, then increases gradually further. After the 40th epoch, the accuracy reaches a plateau and reaches around 92%, indicating that model has learned most of the patterns well.

- **Training Accuracy: 93.8%, Training Loss: 17.9%**
- **Validation Accuracy: 93.3%, Validation Loss: 21.7%**
- **Testing Accuracy: 91.2%, Testing Loss:23.7%**

2.1.2.4.2. Classification Report:

	precision	recall	f1-score	support
AnnualCrop	0.87	0.96	0.91	300
Forest	0.95	0.98	0.97	300
HerbaceousVegetation	0.92	0.91	0.91	300
Highway	0.83	0.88	0.85	250
Industrial	0.97	0.92	0.94	250
Pasture	0.91	0.94	0.93	200
PermanentCrop	0.91	0.88	0.90	250
Residential	0.95	0.99	0.97	300
River	0.96	0.78	0.86	250
SeaLake	0.99	0.97	0.98	300
accuracy			0.92	2700
macro avg	0.93	0.92	0.92	2700
weighted avg	0.93	0.92	0.92	2700

Figure 7: Classification report- CNN

- Precision: High precision means the model makes few false-positive errors. For eg. Precision = 0.99 → When the model predicts "*SeaLake*," it's correct 97% of the time.
- Recall: High recall means the model detects most of the true cases. For eg. Recall = 0.95 → The model identifies 96% of all *AnnualCrop* samples correctly.
- High F1-scores indicate both good precision and recall. For eg. F1-score = 0.87 → The model's overall balance between precision and recall for this class is 87%.
- Support: Number of true samples for each class in the dataset. For eg. For *Residential class*, there are 300 samples in the dataset.
- Accuracy: Overall accuracy of the model is 92%
- Macro Average: Averages the precision, recall, and F1-score across all classes **equally** with a high value of 92%
- **Weighted Average**: Averages the metrics, weighted by the number of true instances (support) in each class.
- **Inference**: Classes *Highway* and *AnnualCrop* have lower precision, recall, and F1-scores (around 0.83-0.87), suggesting the model struggles more with these classes. *Industrial* and *SeaLake* classes have very high F1-scores (0.97 and 0.98), meaning the model performs exceptionally well on these.

2.1.2.4.3. Confusion Matrix:

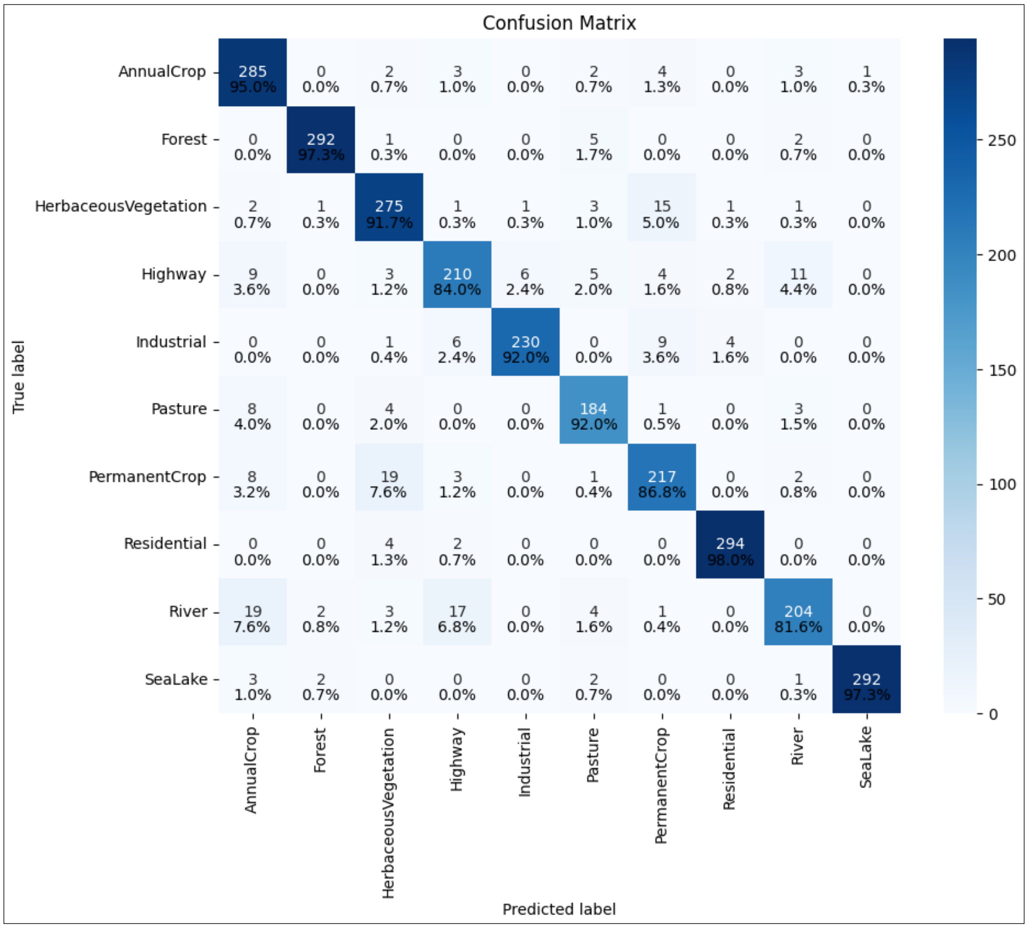

Figure 8: Confusion Matrix - CNN

Inference:

- The diagonal values of the matrix indicate the correctly classified samples for each class. From the diagram, the classifier performs well for most classes, with accuracy above 90% for most.
- "*AnnualCrop*" has 95.0% accuracy (285 correctly classified out of 300 total). "*Residential*" has the highest accuracy at 98.0% (294 out of 300).
- There are a few mismatches, for eg. "*River*" is often misclassified as "*AnnualCrop*" (19 samples) and "*Highway*" (17 samples). "*PermanentCrop*" is frequently misclassified as "*HerbaceousVegetation*" (19 samples).

2.1.2.4.4. ROC Curve:

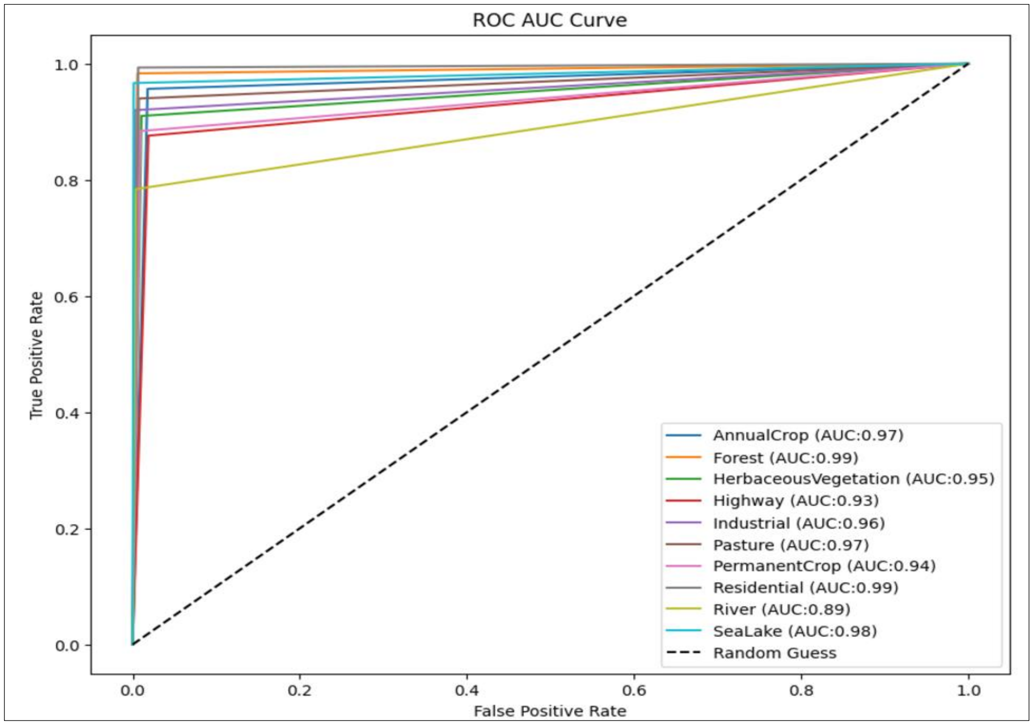

Figure 9: ROC- AUC Curve- CNN

Inference:

- The model performs exceptionally well for most classes, as the AUC values are high (above 0.90 for 9 out of 10 classes).
- "*River*" stands out as the weakest-performing class (AUC = 0.89), consistent with misclassifications observed in the confusion matrix.
- The ROC curves for "*SeaLake*," "*Pasture*," and "*AnnualCrop*" are also strong, but slightly less ideal compared to "Forest" and "Residential."
- Thus, the model is far away from random guessing and performs well on most of the classes.

3. Transfer Learning: Transfer learning using VGG16 is a popular method to leverage a pre-trained model for tasks like image classification. The EuroSAT dataset involves satellite imagery with specific features such as land use patterns, textures, and colors. We choose this particular pretrained model as lower-level features (e.g., edges and textures) learned by VGG16 are highly relevant and can be adapted for the EuroSAT image classification task.

3.1 Process Followed:

3.1.1 Data Preprocessing: Before feeding the images to the neural network, images were preprocessed by implementing the following:

3.1.1.1 Resizing and Normalization: Images are resized to 64×64 pixels for uniform input dimensions. Pixel values are normalized by dividing by 255, scaling them to the range [0, 1]

3.1.1.3 Data Augmentation: Augmentation techniques like rotation, zoom, shear, width/height shifts, and horizontal flipping are applied using ImageDataGenerator to increase dataset diversity.

3.1.1.4 Optimizing Data Loading: Efficient loading and training are ensured by using prefetching and batching to improve data pipeline performance.

3.2 Model Training:

3.2.1 Model Architecture: Following diagram represents the architecture of the CNN model. The input shape is (64,64,3).

Model: "functional"

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 64, 64, 3)	0
block1_conv1 (Conv2D)	(None, 64, 64, 64)	1,792
block1_conv2 (Conv2D)	(None, 64, 64, 64)	36,928
block1_pool (MaxPooling2D)	(None, 32, 32, 64)	0
block2_conv1 (Conv2D)	(None, 32, 32, 128)	73,856
block2_conv2 (Conv2D)	(None, 32, 32, 128)	147,584
block2_pool (MaxPooling2D)	(None, 16, 16, 128)	0
block3_conv1 (Conv2D)	(None, 16, 16, 256)	295,168
block3_conv2 (Conv2D)	(None, 16, 16, 256)	590,080
block3_conv3 (Conv2D)	(None, 16, 16, 256)	590,080
block3_pool (MaxPooling2D)	(None, 8, 8, 256)	0
block4_conv1 (Conv2D)	(None, 8, 8, 512)	1,180,160
block4_conv2 (Conv2D)	(None, 8, 8, 512)	2,359,808
block4_conv3 (Conv2D)	(None, 8, 8, 512)	2,359,808
block4_pool (MaxPooling2D)	(None, 4, 4, 512)	0
block5_conv1 (Conv2D)	(None, 4, 4, 512)	2,359,808
block5_conv2 (Conv2D)	(None, 4, 4, 512)	2,359,808
block5_conv3 (Conv2D)	(None, 4, 4, 512)	2,359,808
block5_pool (MaxPooling2D)	(None, 2, 2, 512)	0
global_average_pooling2d (GlobalAveragePooling2D)	(None, 512)	0
dense (Dense)	(None, 2048)	1,050,624
dense_1 (Dense)	(None, 10)	20,490

Total params: 15,785,802 (60.22 MB)

Trainable params: 1,071,114 (4.09 MB)

Non-trainable params: 14,714,688 (56.13 MB)

Figure 10: VGG16 Architecture

- Used MaxPooling layers throughout the model to reduce the spatial dimensionality of the images after convolutional operations.
- Used `GlobalAveragePooling2D` layer to reduce each feature map to a single number, effectively summarizing spatial information into a feature vector.

3.2.2 Hyperparameters: Following diagram represents the architecture of the CNN model. The input shape is (64,64,3).

- **Activation Functions**
  - **ReLU:** Used in inner layers to replace negative values with zeros, introducing non-linearity and sparsity into the model.
  - **Softmax:** Used in the output layer for multi-class classification.
- **Number of Epochs :** 20 epochs in the code.
- **Batch Size :**32.
- **Early Stopping**
  - Patience: 10 epochs.
- Monitoring: Validation categorical accuracy (`val_categorical_accuracy`).
  - Restores best weights when validation accuracy does not improve.
- **Checkpoints**
  - Used to Save Best Model Weights**:** Saved model weights based on the best validation categorical accuracy (`val_categorical_accuracy`).
- **Optimizer**

  Adam: Used as the optimizer to handle sparse gradients effectively.

3.2.3 Results:

- The following Images represent the training loss (Figure 11) and validation loss (Figure 12), for measuring accuracy, over 20 epochs.

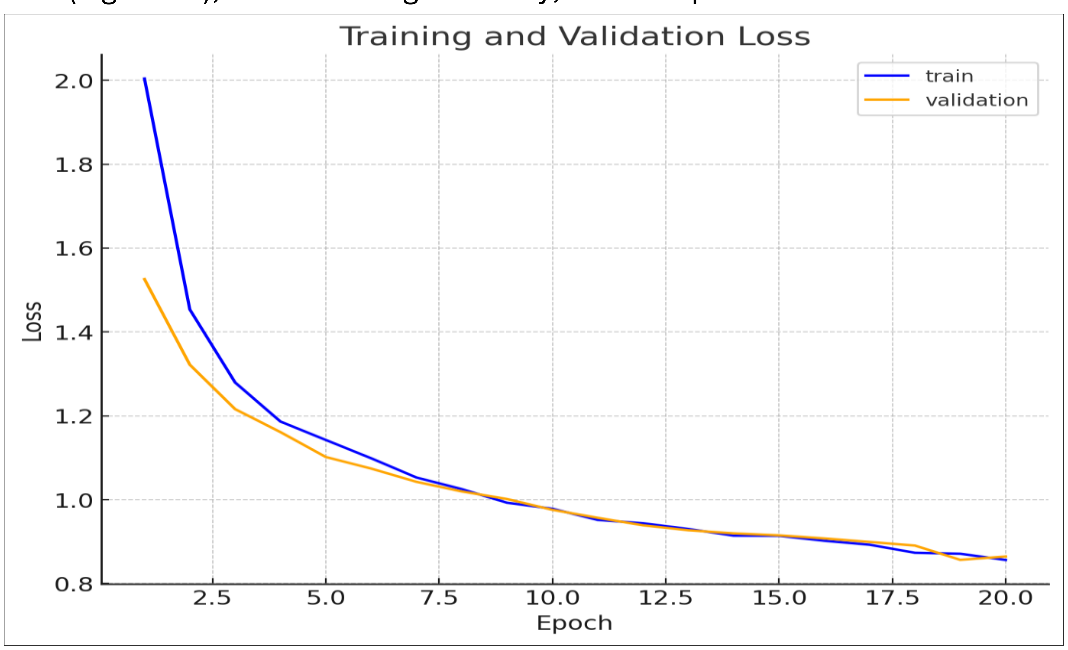

Figure 11: Training/ Validation Loss: VGG16

Inference

- The training loss steadily decreases as the epochs progress, indicating that the model is learning from the training data.
- The validation loss also decreases, showing that the model's performance on unseen data improves over time.
- The close alignment between training and validation loss indicates that the model is not overfitting the training data.
- The gradual convergence of both losses at lower values suggests that the model is well-optimized for the task.
- There are no significant fluctuations in the validation loss, indicating stable training without major generalization issues.

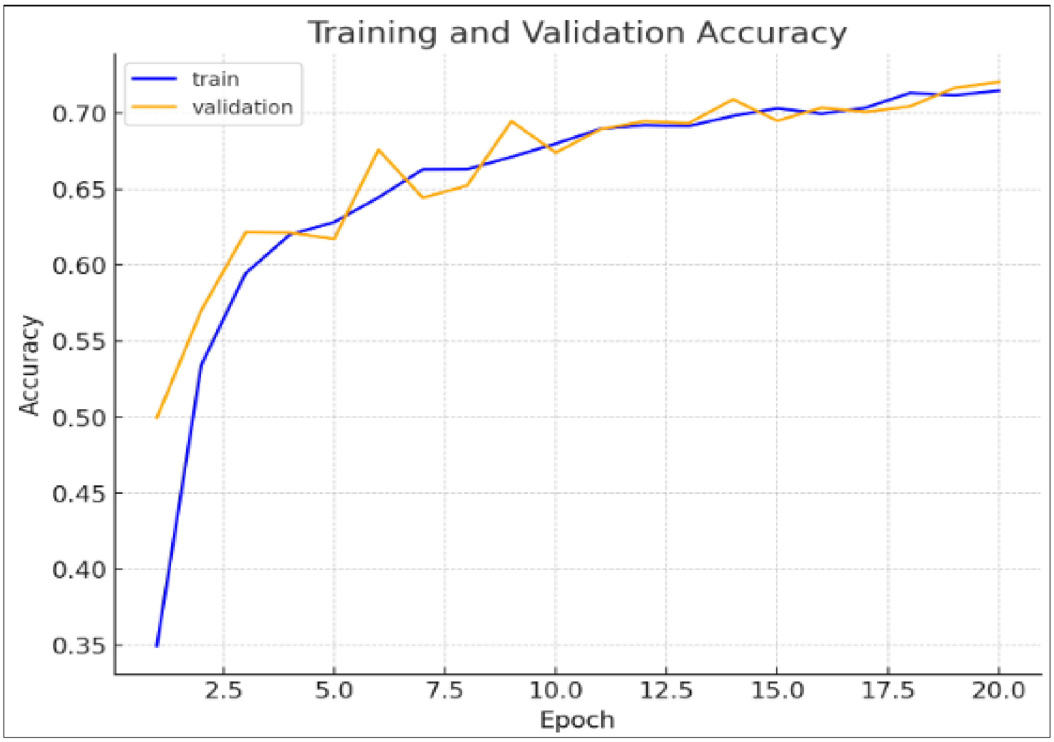

Figure 12: Training/Validation Accuracy: VGG16

Inference -

- Both training and validation accuracy increase steadily, indicating that the model is learning to classify the data effectively.
- The validation accuracy closely follows the training accuracy, showing good generalization to unseen data.

- The training accuracy converges slightly faster than the validation accuracy but remains within a similar range, suggesting minimal overfitting.
- The consistent improvement in accuracy with minor fluctuations in the validation accuracy indicates stable training.
- By the final epochs, the model achieves high accuracy on both training and validation data, implying that it has successfully captured the patterns in the dataset.
  - **Training Accuracy: 71.91%; Training Loss: 42.73%**
  - **Validation Accuracy: 71.44%; Validation Loss: 43.14%**
  - **Test Accuracy: 75.40%**

- **Classification Report**

Classification Report:

	precision	recall	f1-score	support
AnnualCrop	0.820335	0.866000	0.842549	3000.000
Forest	0.791942	0.878000	0.832754	3000.000
HerbaceousVegetation	0.720273	0.704667	0.712384	3000.000
Highway	0.652080	0.602000	0.626040	2500.000
Industrial	0.779162	0.900400	0.835405	2500.000
Pasture	0.640409	0.532500	0.581491	2000.000
PermanentCrop	0.729049	0.567200	0.638020	2500.000
Residential	0.680376	0.965000	0.798070	3000.000
River	0.727800	0.501600	0.593891	2500.000
SeaLake	0.940839	0.874667	0.906547	3000.000
accuracy	0.754000	0.754000	0.754000	0.754
macro avg	0.748226	0.739203	0.736715	27000.000
weighted avg	0.754161	0.754000	0.747159	27000.000

Figure 13: Classification report: VGG16

Inference:

○ The precision, recall, and F1-scores vary across classes. For instance, the model performs well on "SeaLake" (highest F1-score: 0.906) but struggles with "Pasture" (lowest F1-score: 0.581).

○ The model achieves an accuracy of 75.4%, indicating it correctly classifies approximately three-quarters of the test samples.

○ Some classes, like "Industrial" (recall: 0.90), are identified very well, while others, like "Residential" (recall: 0.50), are less accurately identified, suggesting the model may have difficulty with certain class features.

o The average F1-score is 0.736, indicating that performance across all classes is averaged equally. The weighted average F1-score is 0.747, showing the score is slightly better for classes with more samples.

o The model demonstrates moderate performance with an overall accuracy of **75.4%**, which indicates it is reasonably effective but has room for improvement

- **Confusion Matrix:**

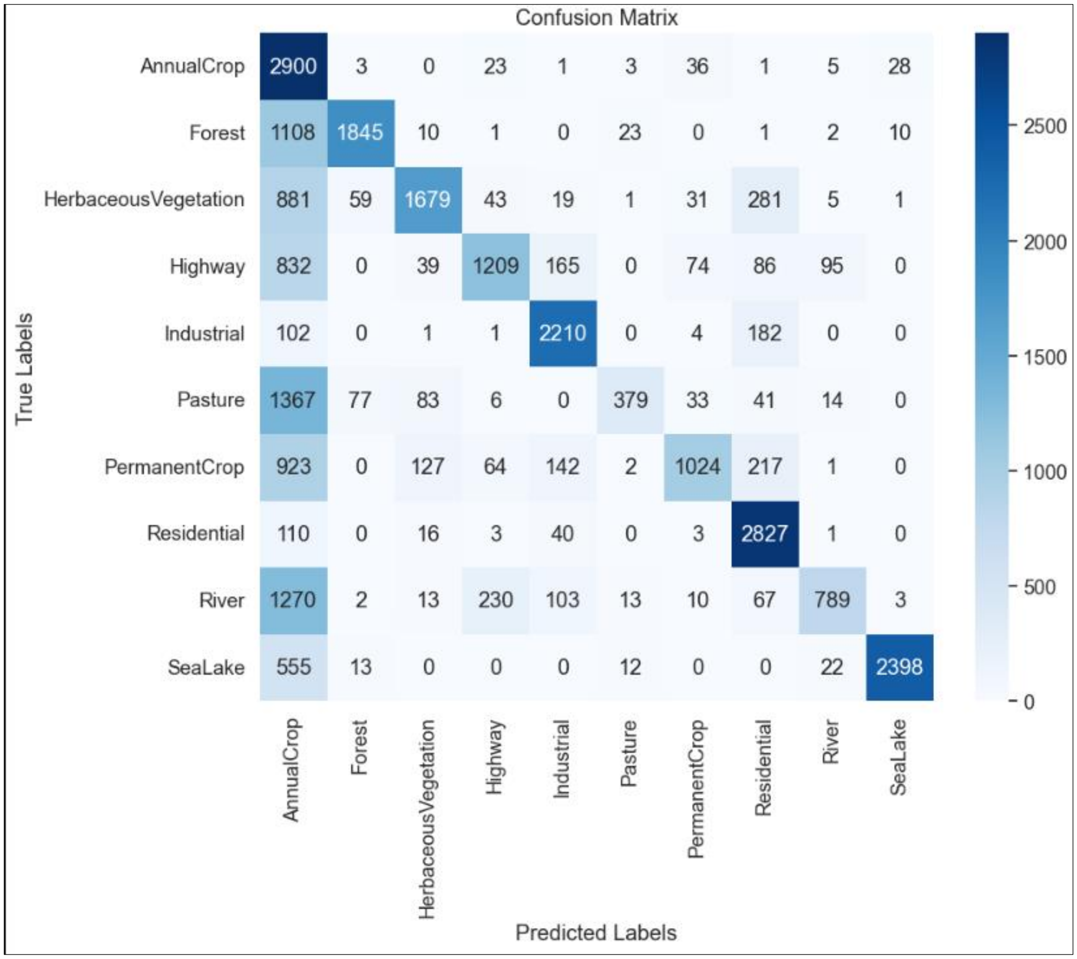

Figure 14: Confusion Matrix: VGG16

Inference:

o Class "AnnualCrop" and "Residential" are well-classified, as most predictions fall into their respective true categories, with minimal misclassifications.

o Class "Pasture" is significantly misclassified, with a large number of samples incorrectly predicted as "AnnualCrop," indicating overlapping features or insufficient representation of "Pasture."

- "SeaLake" shows strong performance, with most samples correctly classified and few misclassifications.
- Some confusion exists between similar land-use classes, such as "HerbaceousVegetation" and "Forest," as well as "River" and "AnnualCrop," suggesting feature similarities in the dataset.
- The model is overall better at distinguishing distinct classes (e.g., "SeaLake") but struggles with subtle differences between certain vegetation or landscape types.

- **ROC Curve**

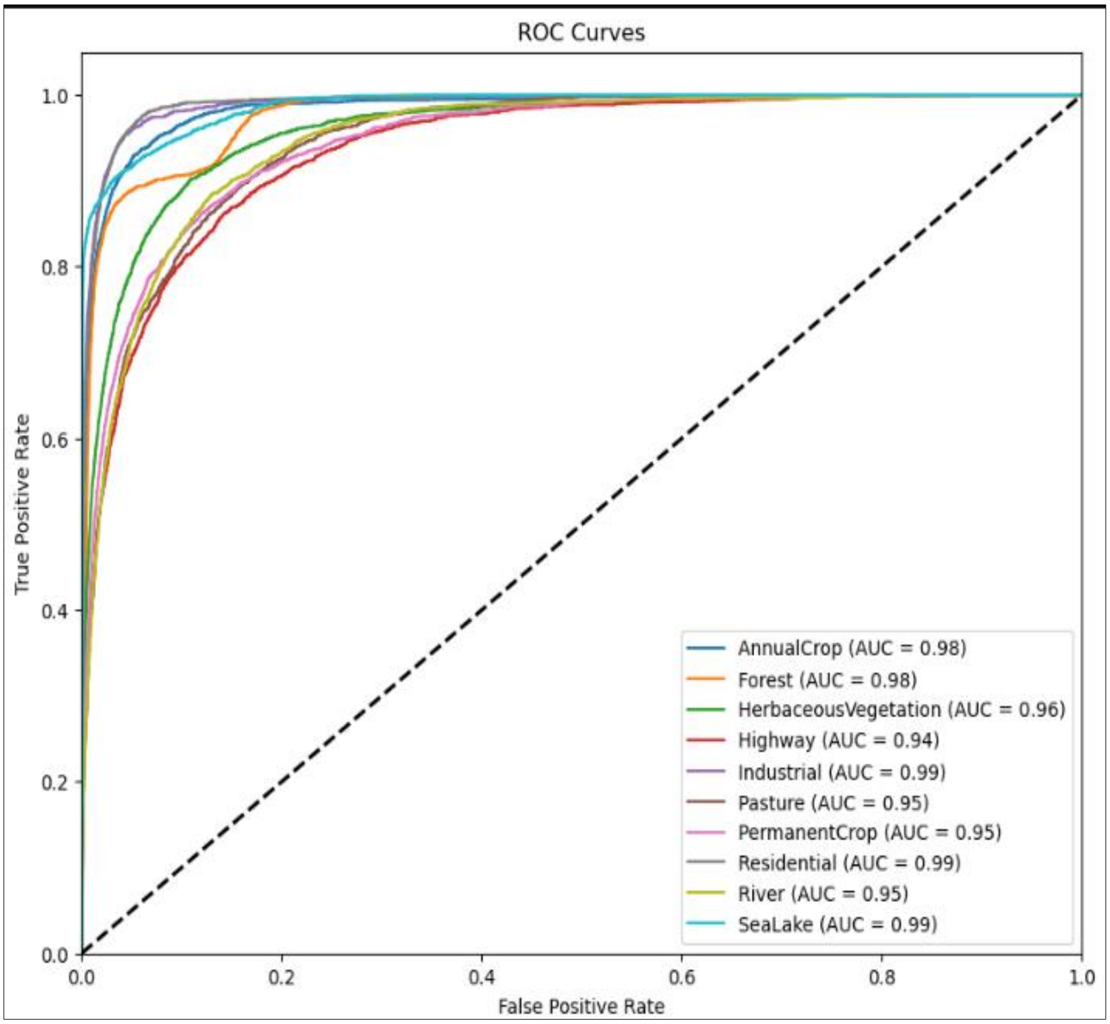

Figure 15: ROC-AUC Curve: VGG16

Inference

- The high AUC values for most classes (close to or above 0.95) indicate that the model has excellent discriminatory power for almost all classes.

- Classes like "Industrial," "Residential," and "SeaLake" achieve near-perfect AUC values (0.99), suggesting the model is highly confident in distinguishing these classes.
- Classes like "Highway" (AUC = 0.94) and "HerbaceousVegetation" (AUC = 0.96) have slightly lower AUCs, implying that the model occasionally struggles to differentiate these classes from others.
- The curves remain close to the top-left corner for most classes, showing a good balance between true positives and false positives across the dataset.
- The overall ROC curves highlight that the model is robust and effective for multi-class classification, with slight room for improvement in distinguishing a few challenging classes.

COMPARATIVE STUDY:

Criteria	SVM	CNN	VGG16
Training Time	19 minutes, 45 seconds	13 minutes	1 hour, 16 minutes, 50 seconds
Accuracy	87.74%	93.00%	75.40%
Precision (Macro Avg)	0.89	0.93	0.75
Recall (Macro Avg)	0.87	0.93	0.74
F1-Score (Macro Avg)	0.88	0.92	0.74
Variance in F1-Scores	Moderate variance across classes (e.g., Forest: 0.88, Pasture: 0.79)	Low variance (e.g., F1-scores mostly above 0.91)	High variance across classes (e.g., SeaLake: 0.91, Pasture: 0.58)
Memory Efficiency	High Memory Consumption	Memory Efficient	Memory Intensive
Model Complexity	Simple preprocessing with SVM	Moderate complexity	High complexity
Scalability	Difficult to scale to larger datasets	Scales well with larger datasets	Scales with computational cost

Generalization	Moderate generalization across classes, relies heavily on preprocessing.	Excellent generalization due to inherent learning capability.	Generalizes well with transfer learning, struggles without pretrained weights.
Key Strengths	Satisfactory with pre-trained preprocessing. Performs well across classes.	High accuracy. No need for complex preprocessing.	Useful for feature extraction with transfer learning. Strong theoretical foundation.
Weaknesses	High memory usage. Slower training compared to CNN.	Relatively shorter training time. Computationally less intensive than VGG16.	Training time is the highest. Accuracy lower than SVM and CNN.
Interpretability	Easy to interpret due to its simplicity.	Moderate interpretability with feature maps visualizable.	Complex to interpret due to deep architecture.
Suitability for Small Datasets	Excellent with proper preprocessing.	Performs moderately well but may require augmentation.	Overfits or underperforms without transfer learning.
Suitability for Large Datasets	Limited due to computational constraints.	Performs well and scales effectively.	Performs well but requires significant computational resources.
Use Case Suitability	Best for tasks with small to medium datasets.	Ideal for general-purpose classification tasks with moderate resources.	Suitable for transfer learning and feature extraction tasks requiring high detail.
Preprocessing Needs	Requires pre-trained models for feature extraction.	Minimal preprocessing with built-in capability.	Requires significant preprocessing or transfer learning.

Table 1: Comparison of Three Models.

CONCLUSION:

In conclusion, the comparative study of SVM, CNN, and VGG16 models highlights that the CNN model is the most effective for classifying land cover images. With an accuracy of 93%, it outperforms SVM (87.74%) and VGG16 (75.40%), demonstrating superior classification capability. The CNN model also exhibits low variance in F1-scores, ensuring consistent performance across all classes, and efficiently handles overlapping features and class imbalances. Moreover, it combines high accuracy with computational efficiency, requiring only 13 minutes of training time, significantly less than SVM and VGG16. Its memory efficiency and excellent generalization capability further enhance its practicality for large-scale classification tasks. Therefore, based on accuracy, consistency, and efficiency, the CNN model is the optimal choice for classifying land cover images.

REFERENCES:

[1] Ziming Li a, Bin Chen a b c, Shengbiao Wu a, Mo Su d, Jing M. Chen e f, Bing Xug;Deep learning for urban land use category classication: A review and experimentalassessment.
[2] Zhe Wang, Chao Fan, Xian Min, Shoukun Sun, Xiaogang Ma, Xiang Que; Cross-scaleUrban Land Cover Mapping: Empowering Classication through Transfer Learning andDeep Learning Integration.
[3] Giorgos Mountrakis, Shahriar S. Heydari; Harvesting the Landsat archive for landcover land use classication using deep neural networks: Comparison with traditionalclassiers and multi-sensor benets.
[4] Ava Vali,Sara Comai,Matteo Matteucci; Deep Learning for Land Use and Land CoverClassication Based on Hyperspectral and Multispectral Earth Observation Data: AReview.
[5] Suraj Sawant, Jayanta Kumar Ghosh; Land use land cover classication usingSentinel imagery based on deep learning models .
[6] Giorgos Mountrakis, Shahriar S. Heydari; Harvesting the Landsat archive for landcover land use classication using deep neural networks: Comparison with traditionalclassiers and multi-sensor benets.
[7] Enhancing land cover classication in remote sensing imagery using an optimal deeplearning model; Abdelwahed Motwake; Aisha Hassan Abdalla Hashim 1 ,Marwa Obayya, Majdy M. Eltahir
[8] Lou, Chen, Mohammed A. A. Al-qaness, Dalal AL-Alimi, Abdelghani Dahou,Mohamed Abd Elaziz, Laith Abualigah, and Ahmed A. Ewees. 2024. “Land Use/LandCover (LULC) Classication Using Hyperspectral Images: A Review.” Geo-SpatialInformation Science