Minor Project-I Report

Land Use Land Cover Classification using Deep Learning

Submitted for Minor Project (CS64123) of
6th semester for partial fulfillment of the
requirements for the award of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Anshumali Suri - 2106024
Ritwik Ramesh Kumar Singh - 2106037
Anshu Gupta - 2106089

Under the guidance of

Dr. Md. Tanwir Udding Haider
Associate Professor, Dept. of CSE

Department of Computer Science & Engineering
National Institute of Technology Patna
Patna, India-800005

May 2024

NATIONAL INSTITUTE OF TECHNOLOGY PATNA

INDIA, 800005

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled **Land Use Land Cover Classification using Deep Learning** submitted by:

Anshumali Suri - 2106024
Ritwik Ramesh Kumar Singh - 2106037
Anshu Gupta - 2106089

in the partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering is an authentic work carried out by them under my supervision and guidance.

Dr. Md. Tanwir Uddin Haider **Date: 02/05/24**

DECLARATION

We, hereby declare that the following report which is being presented in the Minor Project (CS64123) entitled as **Land Use Land Cover Classification using Deep Learning** is an authentic documentation of our own original work to the best of our knowledge. The following project and its report in part or whole, has not been presented or submitted by us for any purpose in any other institute or organization. Any contribution made to the research by others, with whom we have worked at or elsewhere, is explicitly acknowledged in the report.

Anshumali Suri - 2106024

Ritwik Ramesh Kumar Singh - 2106037

Anshu Gupta - 2106089

ACKNOWLEDGEMENT

With due respect, we express our deep sense of gratitude to our respected guide and coordinator **Dr.Md.Tanwir Uddin Haider**, Associate Professor, Department of Computer Science and Technology, for his valuable help and guidance. We are thankful for the encouragement that he has given us in completing this project successfully

It is imperative for us to mention the fact that the report of the Minor Project-I could not have been accomplished without the periodic suggestions and advice of our project guide Dr. Md.Tanwir Uddin Haider.

We are also thankful to all the other faculty, staff members and laboratory attendants of our department for their kind cooperation and help. Last but certainly not the least; we would like to express our deep appreciation towards our family members and batch mates for providing support and encouragement.

Anshumali Suri - 2106024

Ritwik Ramesh Kumar Singh - 2106037

Anshu Gupta - 2106089

Synopsis

Our research project endeavors to develop a streamlined approach to Land Use Land Cover (LULC) Classification using Hyperspectral Remote Sensing Imagery. Our primary aim is to create a simplified model that offers accuracy comparable to state-of-the-art methods, while also addressing critical challenges encountered in this domain. To achieve this, we prioritize the creation of a model that demands less training time without sacrificing accuracy. We employ techniques like feature engineering and model optimization to streamline the process and ensure efficient performance.

In parallel, we tackle issues of class imbalance by integrating the Synthetic Minority Oversampling Technique (SMOTE) into our pipeline, ensuring robust performance across all classes, including underrepresented ones. Furthermore, we leverage domain knowledge through the incorporation of the normalized difference vegetation index (NDVI), enhancing the model's ability to discern between different land cover types and crop species. Additionally, we employ dimensionality reduction techniques, such as Principal Component Analysis (PCA), to optimize performance and reduce training time, thereby preventing overfitting. Through these efforts, we aim to develop an efficient and accurate solution that contributes to sustainable land management practices and informed decision-making processes in diverse socio-economic and environmental contexts.

This project's contributions are manifold.:

1. **Efficient Training**: Novel approach reduces training time while maintaining accuracy, diverging from resource-intensive conventional methods for LULC classification.

2. **Robust Performance**: SMOTE integration ensures reliable classification across diverse classes, addressing common challenges of class imbalance in remote sensing.

3. **Domain Knowledge Integration**: NDVI incorporation enhances model's vegetation type differentiation, advancing accuracy and reliability in crop identification tasks.

4. **Efficiency Through PCA**: Dimensionality reduction via PCA optimizes training time, mitigates overfitting, and boosts generalization in remote sensing applications.

Keywords

land use land cover, deep learning, hyperspectral images, hsi

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Overview

Our project aims to revolutionize land use land cover (LULC) classification and crop identification using hyperspectral remote sensing imagery. Leveraging state-of-the-art deep learning techniques and domain-specific knowledge, we seek to address the challenges inherent in accurately mapping and categorizing various land cover types and crop species. By combining advanced methodologies with innovative approaches, our project endeavors to enhance the efficiency, accuracy, and scalability of existing classification models in the field of precision agriculture, environmental monitoring, and urban planning.

1.2 Objectives

1. **Develop a Simplistic yet Accurate Model**: Our primary objective is to create a deep learning model that balances simplicity with accuracy. By prioritizing model efficiency and training time, we aim to ensure that our solution is accessible and practical for real-world deployment while still delivering performance that rivals state-of-the-art approaches.

2. **Ensure High Accuracy Across All Classes**: We strive to achieve high accuracy not only for dominant land cover classes but also for underrepresented ones. To address class imbalance challenges, we will implement Synthetic Minority Oversampling Technique (SMOTE) and other techniques to ensure robust classification performance across all categories.

3. **Incorporate Domain Knowledge with NDVI**: Recognizing the limitations of existing models in distinguishing between certain vegetation classes, we aim to incorporate domain knowledge, particularly the Normalized Difference Vegetation Index (NDVI), to improve classification accuracy. By leveraging domain-specific insights, we seek to enhance the model's ability to discriminate between different vegetation types accurately.

4. **Optimize Training Efficiency with Dimensionality Reduction**: o streamline the training process and mitigate the risk of overfitting, we will employ Dimensionality Reduction techniques, such as Principal Component Analysis (PCA). By

reducing the dimensionality of the input data while preserving critical information, we aim to accelerate training times and improve the model's generalization capabilities.

5. **Contribute to Sustainable Land Management**: Ultimately, our research endeavors to contribute to sustainable land management practices by providing accurate, efficient, and scalable tools for land cover mapping and crop classification. Through our innovative approach, we seek to facilitate informed decision-making processes in diverse socio-economic and environmental contexts, ultimately fostering more sustainable land management practices.

1.3 Importance

The importance of Land Use and Land Cover (LULC) classification lies in its ability to provide invaluable insights into the dynamics of Earth's surface, enabling us to understand and monitor changes in land cover over time. By categorizing landscapes into distinct classes such as forests, water bodies, and urban areas, LULC classification facilitates the identification of trends like deforestation, urban expansion, and agricultural intensification. This temporal analysis is crucial for evaluating environmental trends and assessing the impacts of human activities and climate change on landscapes.

Moreover, LULC classification plays a pivotal role in supporting informed decision-making processes across various sectors. From urban planning to natural resource management and environmental conservation, LULC data provides stakeholders with the necessary information to make strategic decisions regarding infrastructural development, habitat conservation, water resource management, and disaster risk reduction. By understanding how human activities and environmental factors influence land use patterns within specific regions, policymakers and planners can implement measures to promote sustainable land use practices, minimize environmental degradation, and safeguard ecosystems for future generations.

Chapter 2

Literature Survey

2.1 Review

- *Bhosle and Musande (2019)* **Evaluation of DCNN for LULC Classification and Crop Identification using Hyperspectral Remote Sensing Images** [1]

  In their study, Bhosle and Musande (2019) explored the utilization of deep learning CNN models for land use land cover (LULC) classification and crop identification using hyperspectral remote sensing images. They leveraged Principal Component Analysis (PCA) to effectively reduce the number of spectral bands, thus improving computational efficiency and mitigating the curse of dimensionality. However, the study encountered limitations such as poor accuracy (79.43%) for the study area dataset, mainly attributed to class imbalances in the train-test-split. They suggested implementing stratified k-fold cross-validation to address this issue. Additionally, the absence of data augmentation techniques limited the diversity of the training dataset, potentially affecting model robustness. Furthermore, undefined sampling strategy and patch size hindered reproducibility, suggesting the need for optimizing these parameters for improved model performance. Moreover, the limited resolution of the study area dataset (30 meters) could impact classification accuracy, highlighting the potential benefits of applying super-resolution techniques for enhanced resolution and accuracy. Finally, integrating domain-specific knowledge, such as Vegetation Indexes like Normalized Difference Vegetation Index (NDVI), from spectral bands could further enhance overall accuracy. This comprehensive exploration underscores opportunities for refinement and enhancement in the implementation of deep learning CNN models for LULC classification and crop identification.

- *Carranza-García, García-Gutiérrez, and Riquelme (2019)* **A Framework for Evaluating LULC Classification using CNN** [2]

  Carranza-García et al. (2019) proposed a framework for evaluating land use and land cover (LULC) classification using Convolutional Neural Networks (CNNs) applied to remote sensing (RS) imagery. They highlighted the significance of analyzing LULC for various environmental and social applications, emphasizing the emergence of deep learning models, particularly CNNs, as state-of-the-art solutions for image classification tasks. However, the study identified several areas for improvement, including the need for a standard validation procedure for machine learning models, such as cross-validation, to enhance the comparison of different methods. Additionally, the authors emphasized the importance of deeper exploration into the configuration of the CNN architecture and parameters, suggesting automated approaches for optimization. They also highlighted the necessity of post-processing techniques to refine classification results and the adaptation of the system to handle fused data from multiple sources and sensors. Moreover, considerations for patch size, feature extraction techniques, and treatment of image edges were discussed as crucial factors influencing classification accuracy and robustness.

Despite these challenges, the study demonstrated the superiority of CNNs over other machine learning algorithms for LULC classification, showcasing high accuracy across diverse datasets. The proposed framework also offered advantages such as reduced computation time and utilization of grid-search experimentation for optimal hyperparameter tuning. This comprehensive examination sheds light on the potential and limitations of CNN-based approaches in RS imagery analysis, paving the way for future advancements in LULC classification methodologies.

- *Rasheed and Mahmood (2023)* **A Framework based on DNN for LULC and Rice Crop Classification without using Survey Data** [3]

  Rasheed and Mahmood (2023) proposed a framework based on deep neural networks (DNNs) for land use land cover (LULC) and rice crop classification without relying on survey data. The study addressed the critical need for precise mapping of crop and land cover types for food security, precision agriculture, and water management. By leveraging machine learning (ML) and deep learning (DL) algorithms, the framework utilized Sentinel-2 time-series satellite datasets to identify rice crop and other land cover types. The normalized difference vegetation index (NDVI) stack, developed through Google Earth Engine (GEE), served as input features for classification. The study evaluated various ML algorithms and DL models, demonstrating the superior performance of DL methods, particularly the Swin Transformer (ST) model, in achieving high classification accuracy.

  However, the study identified challenges related to the differentiation of certain land cover classes with similar vegetation properties, such as Rangeland and Trees. To address this, the authors suggested exploring alternative vegetation indices or combining multiple indices to enhance classification accuracy. Additionally, they recommended the application of data augmentation techniques to further improve model performance. Despite these challenges, the study underscored the significance of accurate LULC classification for informing decision-making processes and planning initiatives, emphasizing the potential of DL-based approaches in extracting valuable insights from satellite imagery without the need for ground survey.

- *Vali A, Comai S, Matteucci M* **Deep Learning for Land Use and Land Cover Classifi- cation Based on Hyperspectral and Multispectral Earth Observation Data: A Review** [4]

  This pivotal paper shed light on the burgeoning interest within the remote sensing community in leveraging deep learning for land use and land cover (LULC) classification, particularly with multispectral and hyperspectral images. Its thorough review of the literature highlighted the exponential growth in related publications since 2015, reflecting the increasing adoption of deep learning techniques for handling the complexities of remote sensing data. Despite the promise of deep learning, the paper also underscored challenges related to ground-truth data, resolution, and data nature, which significantly impact classification performance.

Furthermore, the paper provided invaluable insights into available data sources, datasets, methodologies, and challenges in the field of LULC classification using deep learning. Its framework facilitated a deeper understanding of the state-of-the-art techniques and paved the way for our project's conceptualization and implementation. Many of our project's foundational ideas and approaches find their origins in the methodologies and findings outlined in this seminal paper.

- *H. Ghanbari, M. Mahdianpari, S. Homayouni and F. Mohammadimanesh* **A Meta-Analysis of Convolutional Neural Networks for Remote Sensing Applications** [5]

  This paper presents a meta-analysis of 416 peer-reviewed journal articles, exploring the utilization of convolutional neural networks (CNNs) in remote sensing (RS) applications. It outlines the transition of CNNs as a dominant force in RS, surpassing traditional machine learning methods. The analysis covers various aspects such as application domains, sensor types, and algorithm specifications. By summarizing trends and advancements, it offers valuable insights for researchers exploring CNN-based approaches in RS tasks like land use land cover (LULC) classification.

  This paper's comprehensive review of CNN applications in remote sensing convinced us of its efficacy as a foundational architecture for our project. Its analysis of CNN advancements and successes across diverse RS applications affirmed our decision to adopt CNNs as the cornerstone of our simplified model.

- *Prasad S Thenkabail, Ronald B Smith, Eddy De Pauw* **Hyperspectral Vegetation Indices and Their Relationships with Agricultural Crop Characteristics** [6]

  The study utilized ground-level hyperspectral reflectance measurements of cotton, potato, soybeans, corn, and sunflower, with reflectance data collected in 490 discrete narrow bands between 350 and 1,050 nm. Three types of hyperspectral predictors were examined: optimum multiple narrow band reflectance (OMNBR), narrow band normalized difference vegetation index (NDVI), and soil-adjusted vegetation indices.

  The study revealed that OMNBR models, although effective, faced challenges related to overfitting, which was mitigated by comparing R2 values of crop variables with those computed for random data of a large sample size. Furthermore, a rigorous search procedure identified the best narrow band NDVI predictors of crop biophysical variables. Special narrow band lambda (1) versus lambda (2) plots illustrated effective wavelength combinations and bandwidths for predicting crop biophysical quantities. The study recommended a 12 narrow band sensor, spanning the 350 nm to 1,050 nm range of the spectrum, for optimal estimation of agricultural crop biophysical information. These findings provide valuable insights into selecting appropriate bands for NDVI calculation in our research project.

2.2 Discussion

The literature surveys conducted provide valuable insights into the application of deep learning models for land use land cover (LULC) classification and crop identification using remote sensing imagery. Bhosle and Musande (2019) highlighted the effectiveness of convolutional neural networks (CNNs) in achieving high accuracy for LULC classification tasks, albeit with challenges related to class imbalances and domain-specific knowledge integration. Similarly, Carranza-García et al. (2019) proposed a CNN-based framework for LULC classification, emphasizing the need for standard validation procedures and post-processing techniques to enhance overall results. Furthermore, Rasheed and Mahmood (2023) demonstrated the superiority of DL methods in LULC classification, while identifying challenges related to class differentiation and the exploration of alternative vegetation indices.

In light of these findings, our project incorporates strategies such as SMOTE for addressing class underrepresentation, integration of NDVI for enhancing domain-specific knowledge, utilization of k-fold stratified cross-validation for robust model evaluation, and dimensionality reduction techniques to streamline model complexity and training time. These approaches are essential for improving the accuracy and generalization capabilities of our model, aligning with the advancements and insights gleaned from existing research in the field.

Chapter 3

Methodology

3.1 Workflow

1. **Data Acquisition & Transformation**: Raw Indian Pines dataset (145x145x200) from Kaggle is transformed into a dataframe (21025x201), with each row representing spectral data for a pixel and the final column denoting class labels.

2. **Normalization & NDVI Integration**: MinMax normalization is applied, followed by the incorporation of NDVI values computed from bands 44 and 29. These values are added as an extra feature before class labels.

3. **Dimensionality Reduction**: PCA is executed to reduce dimensionality. Initially, a plot of cumulative explained variance guides the selection of 40 principal components from the 201 bands. The dataset is then reshaped into an image format (145x145x40), enabling streamlined analysis and visualization.

4. **Patch Extraction**: Patches of size 5x5 centered around each pixel are extracted from the 145x145x40 image. These patches capture local spatial neighborhoods, facilitating the analysis of local spectral information by the CNN model.

5. **Data Augmentation**: Following patch extraction, data augmentation techniques are applied to diversify and enhance the dataset's robustness. Each patch is rotated by 45 degrees seven times, resulting in eight rotated versions of each patch. This augmentation strategy improves the model's ability to generalize and recognize patterns effectively. After patch extraction, rotation, and edge padding, the input data is reshaped into a tensor of shape (168200, 5, 5, 40).

6. **Stratified K-fold Cross-Validation**: Employed stratified k-fold cross-validation, a robust technique for estimating model performance on unseen data. The dataset is divided into k subsets (folds), where the model is trained on k-1 folds and evaluated on the remaining fold. This process repeats k times, ensuring each fold serves as the test set once.

7. **Addressing Class Imbalance**: Applied the SMOTE (Synthetic Minority Oversampling Technique) algorithm to tackle class imbalance, particularly for minority classes. This technique generates synthetic samples for minority classes for

each training fold, thereby improving the model's ability to learn from imbalanced datasets.

8. **Repetitive Training**: Trained the model for multiple repetitions to obtain stable and reliable performance estimates, enhancing the robustness of the model evaluation process.

3.2 Framework

To develop this project our proposed framework is shown in the figure 3.1.

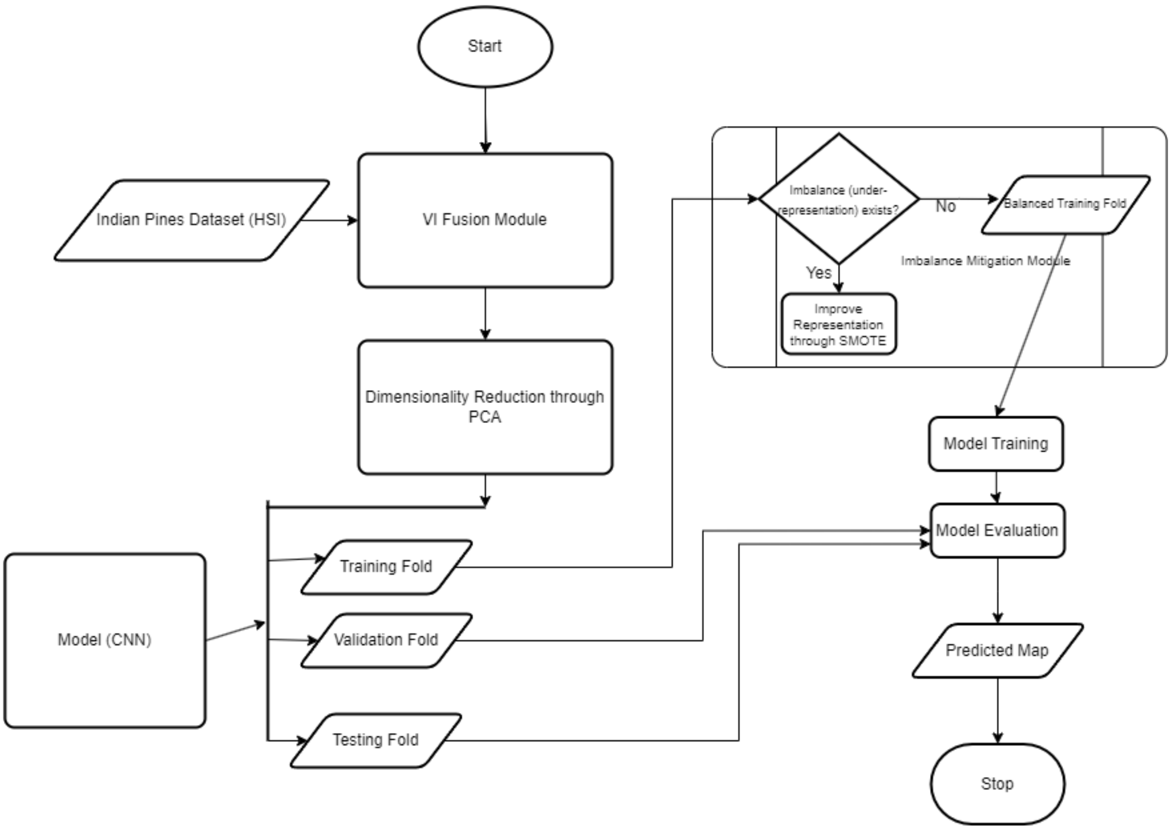

Figure 3.1: The Proposed Framework

3.3 Steps

This framework consists of following steps shown in Figure 3.1. Each of these steps are explained in detail.

3.3.1 Dataset

The dataset "Indian Pines," sourced from Kaggle, comprises a single image containing 200 spectral bands. In Figure 3.2, the image is represented with a distinct band that corresponds to wavelengths captured differently. Once converted into a dataframe, it contains 201 columns and 21,025 entries. Each entry corresponds to the complete spectral information of a specific pixel, as illustrated in Figure 3.3.

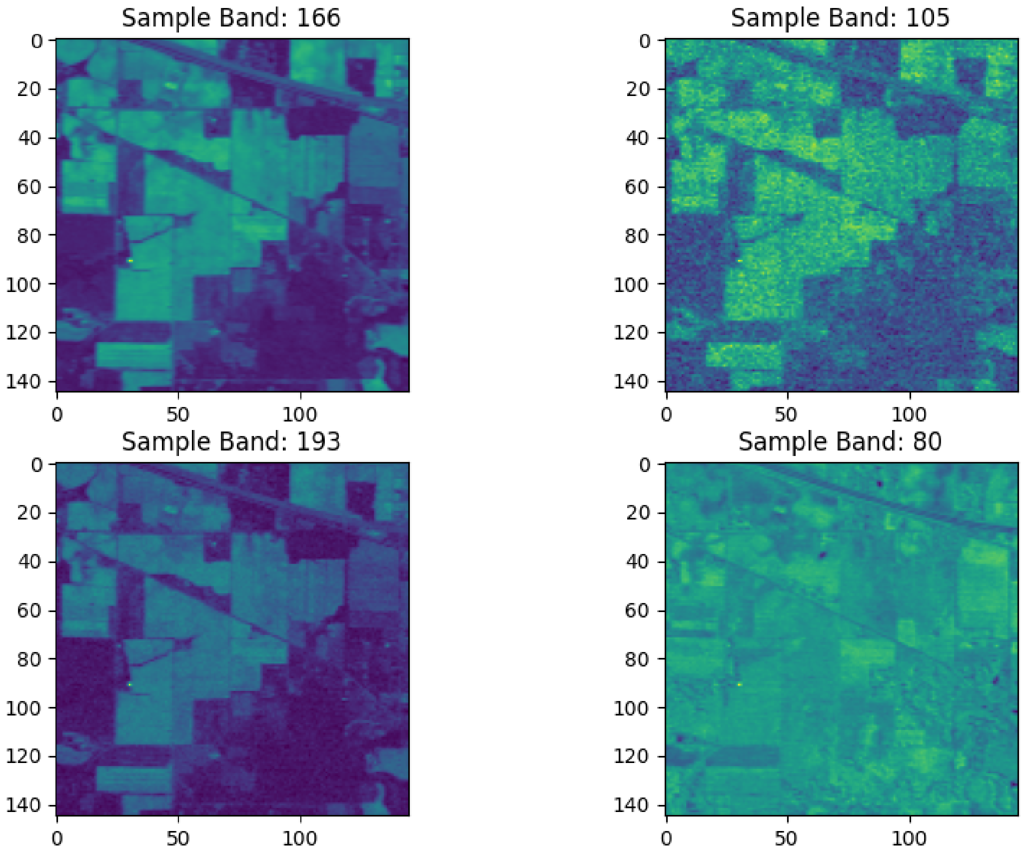

Figure 3.2: Indian Pines Dataset - Band Visualization

	band-1	band-2	band-3	band-4	band-5	band-6	band-7	band-8	band-9	band-10	...	band-192	band-193	band-194	band-195	band-196	band-197	band-198	band-199	band-200	class
21020	2561	3987	4011	4023	4201	4377	4418	4248	4180	3838	...	1013	1012	1018	1015	1011	1001	1000	1009	1008	0
21021	2726	4104	4024	3880	4210	4377	4413	4174	4229	3900	...	1012	1014	1012	1024	998	1010	1006	1000	1000	0
21022	3153	3864	4282	3889	4310	4372	4375	4208	4096	3878	...	1016	1015	1016	1021	1008	1019	1003	1008	1000	0
21023	3155	4104	4106	4027	4139	4318	4413	4174	4140	3933	...	1005	1011	1008	1012	1014	1007	1011	1005	1003	0
21024	3323	3860	4197	3952	4148	4279	4375	4225	3988	3866	...	1018	1014	1007	1015	1002	1010	1007	1004	1000	0

5 rows × 201 columns

Figure 3.3: Snapshot of data before preprocessing

3.3.2 Data Normalization

Min-Max normalization, also known as feature scaling, is a fundamental preprocessing technique used to rescale numerical features within a specific range, typically between 0 and 1. In the project, we applied Min-Max normalization to the spectral bands of the Indian Pines dataset, whose results are depicted in Figure 3.4. This normalization ensures that each feature contributes proportionally to the model's training process, preventing features with larger magnitudes from dominating the learning process. By scaling the data to a consistent range, Min-Max normalization enhances the convergence of optimization algorithms and improves the overall stability and performance of our machine learning models.

	band-1	band-2	band-3	band-4	band-5	band-6	band-7	band-8	band-9	band-10	...	band-192	band-193	band-194	band-195	band-196	band-197	band-198	band-199
20777	0.215081	0.378583	0.068215	0.278435	0.034410	0.041539	0.051636	0.060755	0.026929	0.192562	...	0.069204	0.055172	0.056962	0.037736	0.094787	0.090909	0.101124	0.200000
9089	0.322368	0.419440	0.278392	0.409065	0.259885	0.204638	0.223832	0.267886	0.252546	0.350662	...	0.304498	0.286207	0.287975	0.313208	0.336493	0.272727	0.308989	0.355556
13187	0.315283	0.459638	0.140118	0.324043	0.130999	0.122834	0.127804	0.130659	0.128536	0.243964	...	0.138408	0.086207	0.091772	0.094340	0.161137	0.102767	0.123596	0.288889
11404	0.578947	0.546952	0.442847	0.554899	0.441896	0.424822	0.402804	0.435140	0.439013	0.514603	...	0.415225	0.396552	0.398734	0.377358	0.398104	0.343874	0.398876	0.422222
18257	0.083502	0.458979	0.270649	0.450169	0.260791	0.257136	0.253037	0.262022	0.276759	0.393302	...	0.339100	0.331034	0.325949	0.369811	0.341232	0.335968	0.415730	0.322222

5 rows × 201 columns

Figure 3.4: Dataset After Normalization

3.3.3 Exploratory Data Analysis

In our Exploratory Data Analysis (EDA), we delved into understanding the class distribution within the dataset, a critical step in any machine learning project. By examining the class ratio of the dataset, we gained insights into the distribution of different classes and their respective frequencies. This analysis helps us identify potential class imbalances or biases that could impact the performance of our machine learning models. Additionally, understanding the class distribution guides us in selecting appropriate evaluation metrics and developing strategies to handle class imbalance, such as data augmentation or resampling techniques. Overall, exploring the class ratio during EDA provides valuable insights that inform subsequent steps in our machine learning pipeline, ultimately contributing to the development of more robust and accurate models. The class ratio is depicted in Figure 3.5

3.3.4 NDVI Integration Module

- The NDVI Integration Module serves as a crucial component in our system, leveraging domain knowledge to enhance the representation of spectral information from remote sensing imagery. NDVI, or Normalized Difference Vegetation Index, is a widely used vegetation index that quantifies the presence of vegetation based on the difference in reflectance between near-infrared (NIR) and red bands of the electromagnetic spectrum. The need for NDVI arises from its ability to capture vegetation dynamics and health, making it invaluable for land use and land cover classification tasks.

- In our analysis, we identified that the NIR band corresponds to band-44, while the red band corresponds to band-29 in the spectral bands of the Indian Pines dataset.

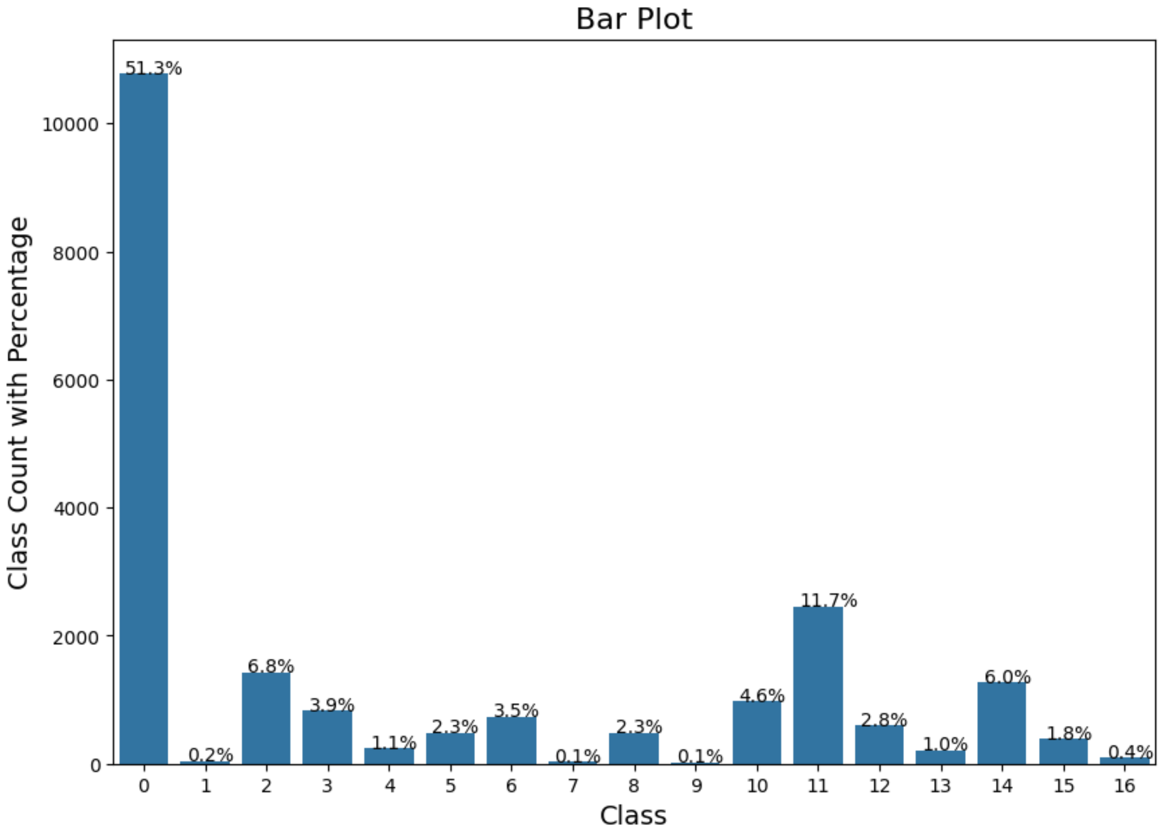

Figure 3.5: Class Count % of Indian Pines Dataset

The NDVI is computed using the following formula:

$$\text{NDVI} = \frac{\text{NIR} - \text{Red}}{\text{NIR} + \text{Red}}$$

- By integrating NDVI into our dataset, we introduce a derived feature that encapsulates valuable information about vegetation density and health. This additional feature augments the spectral data with meaningful insights, contributing to the discriminative power of our machine learning models. Furthermore, appending NDVI at the end of the dataset, as is illustrated in Figure 3.6, ensures that it is readily available for downstream analysis and model training processes, facilitating comprehensive feature representation and improving the overall performance of our system.

band-2	band-3	band-4	band-5	band-6	band-7	band-8	band-9	band-10	...	band-193	band-194	band-195	band-196	band-197	band-198	band-199	band-200	NDVI	class
0.374629	0.135693	0.301802	0.093269	0.069062	0.065888	0.060052	0.052501	0.187695	...	0.031034	0.031646	0.060377	0.090047	0.023715	0.123596	0.200000	0.418182	0.933305	14
0.515980	0.320428	0.435811	0.252641	0.242864	0.264019	0.262960	0.272007	0.368769	...	0.231034	0.227848	0.267925	0.279621	0.213439	0.252809	0.311111	0.527273	0.025879	0
0.422405	0.144174	0.282095	0.111681	0.069572	0.120561	0.112831	0.102059	0.238902	...	0.103448	0.075949	0.135849	0.094787	0.106719	0.151685	0.155556	0.418182	0.624954	6
0.334102	0.067847	0.280405	0.052520	0.042813	0.018224	0.040816	0.040959	0.163941	...	0.044828	0.025316	0.064151	0.056872	0.075099	0.078652	0.255556	0.345455	0.917375	0
0.305437	0.149336	0.288288	0.117114	0.075433	0.100935	0.077645	0.078751	0.209696	...	0.055172	0.028481	0.075472	0.075829	0.071146	0.123596	0.222222	0.600000	0.818556	0
0.464580	0.377950	0.492117	0.352852	0.325178	0.352103	0.350457	0.343290	0.467095	...	0.420690	0.417722	0.426415	0.473934	0.403162	0.471910	0.633333	0.600000	-0.109420	3
0.344975	0.181785	0.324324	0.094174	0.084098	0.097897	0.072250	0.062684	0.195288	...	0.065517	0.044304	0.037736	0.113744	0.051383	0.123596	0.222222	0.436364	0.903645	14
0.430972	0.382006	0.449887	0.332629	0.298930	0.298832	0.319259	0.312740	0.420171	...	0.258621	0.224684	0.256604	0.241706	0.249012	0.247191	0.333333	0.527273	0.121384	0
0.424382	0.114676	0.279842	0.057350	0.055810	0.064252	0.073188	0.055442	0.187500	...	0.027586	0.072785	0.064151	0.090047	0.071146	0.073034	0.266667	0.345455	0.909436	14
0.429654	0.115044	0.305462	0.117718	0.103466	0.114953	0.113066	0.108622	0.232477	...	0.124138	0.104430	0.135849	0.151659	0.142292	0.157303	0.322222	0.345455	0.632641	15

Figure 3.6: Dataset After NDVI Integration

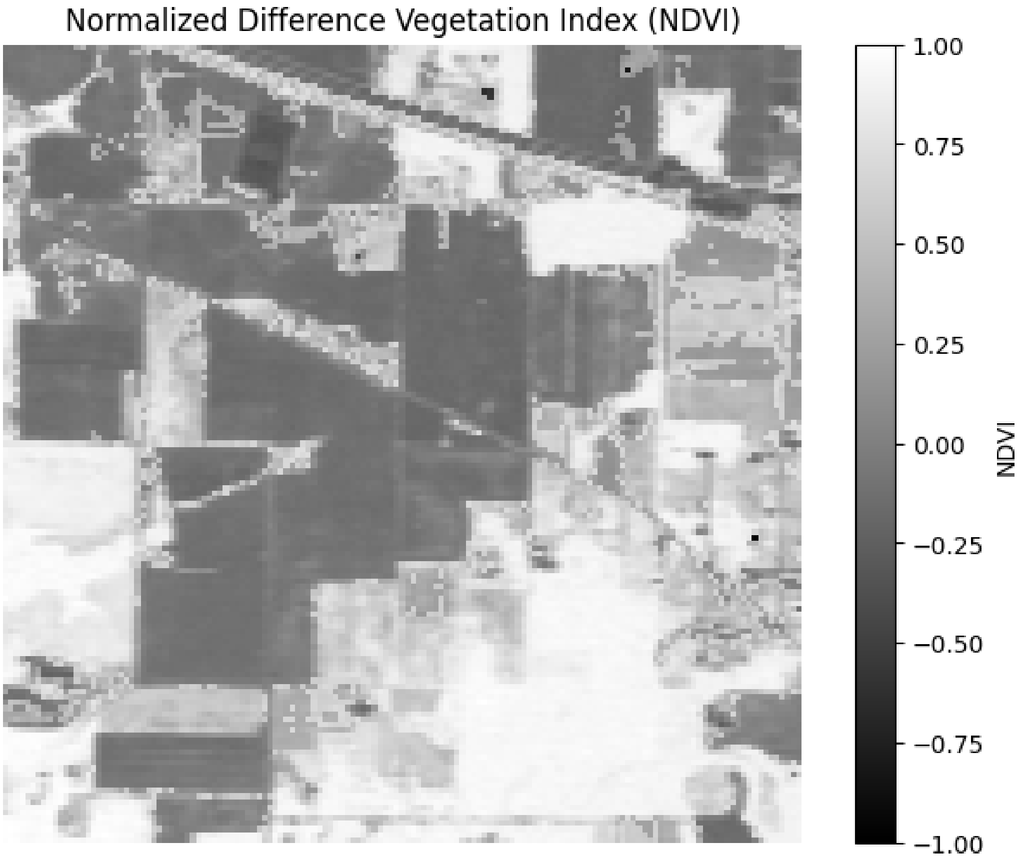

Figure 3.7: NDVI Band Visualization

3.3.5 Dimensionality Reduction Module

- Dimensionality Reduction is a crucial technique employed in our project to streamline the computational complexity and enhance the efficiency of our machine learning pipeline. The primary motivation behind utilizing dimensionality reduction is twofold: to shorten the training time and to eliminate redundancies inherent in the sheer amount of band information present in hyperspectral imagery.

- By reducing the dimensionality of our dataset, we aim to mitigate the computational burden associated with processing high-dimensional data, thereby accelerating the training process of our machine learning models. Additionally, dimensionality reduction enables us to identify and extract the most informative features while discarding redundant or irrelevant ones, leading to more concise and interpretable representations of the data.

- In our methodology, we initiated the process by implementing Principal Component Analysis (PCA) to identify the principal components that capture the maximum variance within the minimum number of dimensions. Figure 3.8 illustrates the cumulative variance contained in the number of principal components. It is apparent that beyond the 40th principal component, the cumulative variance saturates. Consequently, we concluded that retaining 40 principal components adequately preserves the majority of the variance present in the original dataset. Through the application of PCA transformation, we successfully reduced the dimensionality of our dataset from 201 dimensions to 40 dimensions, thereby achieving a significant reduction in computational complexity without compromising essential information.

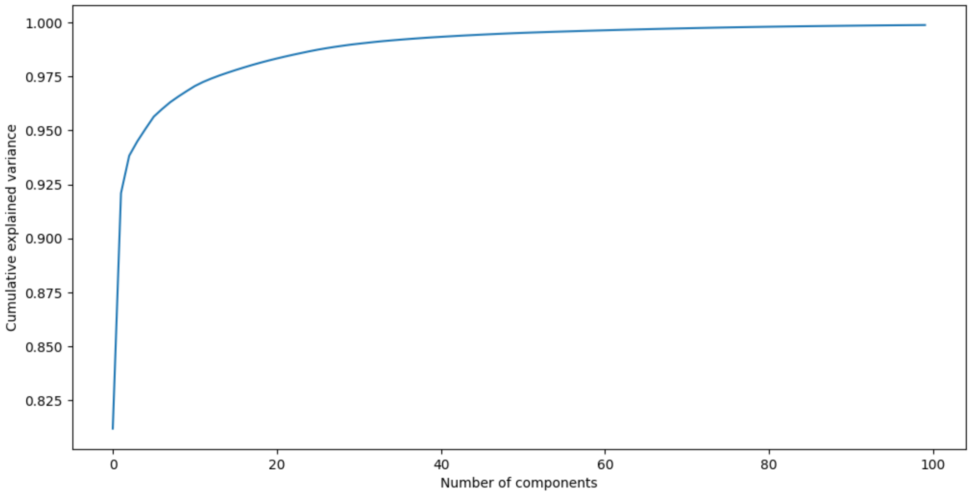

Figure 3.8: Graph displaying the distribution of variance along components

- This reduction in dimensionality not only expedites the training process of our machine learning models but also improves their generalization capabilities by minimizing the risk of overfitting to noise or irrelevant features. Consequently, our streamlined and optimized dataset with reduced dimensions facilitates more efficient and accurate model training, paving the way for enhanced performance and scalability in our project. Figure 3.9 depicts the dataset after application of PCA, where the bands have been reduced to 40.

	PC-1	PC-2	PC-3	PC-4	PC-5	PC-6	PC-7	PC-8	PC-9	PC-10	...	PC-32	PC-33	PC-34	PC-35	PC-36	PC-37	PC-38
17845	0.325834	1.976317	-0.395021	0.296582	-0.057011	-0.079875	0.015338	0.120412	-0.026658	-0.314165	...	-0.015777	0.019628	0.069879	0.039005	0.031044	0.032036	-0.026716
20669	-2.627908	0.194497	0.050112	0.230192	-0.087979	-0.173960	0.045694	-0.046817	0.135515	0.066079	...	0.016853	0.005434	-0.041254	0.002716	-0.007038	-0.001930	-0.013569
11600	-2.611238	-1.216313	-0.002693	0.070901	-0.212133	-0.155757	-0.012474	0.208353	0.075344	0.002491	...	-0.031193	-0.014173	0.041627	-0.008281	-0.018749	0.000901	-0.022128
7269	3.342118	-1.845932	0.441556	-0.041246	0.176427	-0.370893	-0.184368	-0.161019	-0.199210	0.144493	...	0.041697	0.007037	-0.115798	0.131004	-0.034465	-0.021702	0.016139
18301	2.904526	-0.903674	0.052285	-0.202049	-0.192808	-0.120924	0.222554	-0.024542	-0.082654	-0.111168	...	-0.029151	0.017182	-0.106840	0.075605	0.011646	-0.002265	-0.028441
6869	3.129983	-1.202831	0.327626	0.506112	0.369984	-0.070370	-0.039847	-0.180424	-0.006869	-0.059423	...	-0.001511	0.021937	-0.013645	0.086114	-0.018254	0.003620	0.030904

6 rows × 41 columns

Figure 3.9: Dataset After Applying PCA

3.3.6 Data Preparation Steps

- The data preparation steps for our project involve several crucial processes to enhance the quality and diversity of our dataset, ensuring optimal performance of our convolutional neural network (CNN) models.

- Firstly, the original hyperspectral image, sized 145x145x40, is transformed to accommodate the limitations of CNNs in processing high-dimensional data efficiently. Instead of passing the entire image as a single input, we adopt a patch-based approach for feature extraction.

- **Patch Extraction**: Patches of size 5x5 centered around each pixel are extracted from the hyperspectral image. Each patch captures a spatial neighborhood around the pixel, facilitating CNN's analysis of local spectral information. This approach enables the network to focus on small spatial regions, enhancing its ability to detect patterns and features effectively.

- **Data Augmentation**: To enrich the diversity and robustness of our dataset, we apply data augmentation techniques post-patch extraction. Specifically, each patch undergoes rotation by 45 degrees in seven different directions, resulting in eight rotated versions of each patch. This augmentation strategy enhances the dataset's variability, aiding the CNN in generalizing and recognizing patterns more effectively across different orientations.

- **Edge Padding**: An edge padding operation is performed to address edge effects and ensure that patches extracted near the image boundary contain sufficient spatial context. This step enhances the consistency and completeness of the spatial information captured by the patches, mitigating potential biases introduced by boundary pixels.

Overall, these data preparation steps culminate in an augmented dataset comprising patches with enhanced diversity and spatial context. With a final dataset shape of (168200, 5, 5, 40) [ref Fig. 3.10], our input data is well-prepared to feed into CNN architectures for subsequent training and analysis, enabling robust and accurate hyperspectral image classification.

```
((168200, 5, 5, 40), (168200,))
```

Figure 3.10: Dataset Shape After Data Preparation Steps

3.3.7 Model Architecture

The model architecture, as depicted in Figure 3.11, is designed to effectively process and classify hyperspectral image patches for land cover classification tasks. It consists of several layers arranged sequentially to extract features and make predictions.

1. **Convolulational Layers**:

    - Conv2D (32 filters, kernel size 3x3, ReLU activation, padding='same'): The first convolutional layer processes the input data, which has a shape of (5, 5, 40). It applies 32 filters to extract spatial features, followed by the ReLU activation function to introduce non-linearity. The padding parameter is set to 'same' to preserve spatial dimensions.

    - BatchNormalization: Batch normalization is applied after each convolutional layer to normalize the activations, improving the stability and convergence of the model.

- MaxPooling2D (pool size 2x2): Max pooling is performed to downsample the feature maps, reducing their spatial dimensions while retaining the most relevant information.

2. **Flattening and Dense Layers**:

   - Conv2D (64 filters, kernel size 3x3, ReLU activation, padding='same'): The second convolutional layer further processes the feature maps generated by the previous layers, increasing the depth of feature representation.
   - BatchNormalization: Similar to the first batch normalization layer, this layer normalizes the activations to stabilize the learning process.
   - MaxPooling2D (pool size 2x2): Another max pooling operation is applied to downsample the feature maps before passing them to the fully connected layers.
   - Flatten: The Flatten layer reshapes the 3D feature maps into a 1D vector, preparing them for input to the dense layers.
   - Dense (1024 units, ReLU activation): The flattened feature vector is passed through a dense layer with 1024 units, introducing additional non-linearity and complexity to the model.
   - Dropout (rate=0.2): Dropout regularization is applied to the dense layer to prevent overfitting by randomly dropping a fraction of the units during training.
   - Dense (17 units, softmax activation): The final dense layer consists of 17 units, corresponding to the number of land cover classes in the dataset. It applies the softmax activation function to compute the probability distribution over the classes, enabling multi-class classification.

3. **Model Compilation**:

   - Loss Function: The model is trained using the sparse categorical cross-entropy loss function, which is suitable for multi-class classification tasks with integer labels.
   - Optimizer: Stochastic Gradient Descent (SGD) optimizer is utilized to minimize the loss function. The learning rate is scheduled using an exponential decay function to adaptively adjust the learning rate throughout training, improving convergence and stability.
   - Metrics: The accuracy metric is used to evaluate the performance of the model during training and validation.

The resulting model architecture is well-suited for processing hyperspectral image patches and effectively classifying land cover types, offering a balance between model complexity and computational efficiency.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 5, 5, 32)          11552

 batch_normalization (Batch  (None, 5, 5, 32)          128
 Normalization)

 max_pooling2d (MaxPooling2  (None, 2, 2, 32)          0
 D)

 conv2d_1 (Conv2D)           (None, 2, 2, 64)          18496

 batch_normalization_1 (Bat  (None, 2, 2, 64)          256
 chNormalization)

 max_pooling2d_1 (MaxPoolin  (None, 1, 1, 64)          0
 g2D)

 flatten (Flatten)           (None, 64)                0

 dense (Dense)               (None, 1024)              66560

 dropout (Dropout)           (None, 1024)              0

 dense_1 (Dense)             (None, 17)                17425

=================================================================
Total params: 114417 (446.94 KB)
Trainable params: 114225 (446.19 KB)
Non-trainable params: 192 (768.00 Byte)
_________________________________________________________________
```

Figure 3.11: Model Architecture

3.3.8 Training Methodology

- **Validation Strategy Rationale**:
  - Stratified K-Fold Cross-Validation: Stratified K-Fold cross-validation is chosen to mitigate the risk of bias introduced by class imbalance. By ensuring that each fold preserves the same class distribution as the original dataset, this technique provides a more reliable estimate of model performance across different subsets of the data.
  - Reason for Stratification: Stratification is crucial in scenarios where the class distribution is uneven. Without it, there's a risk of some classes being underrepresented or entirely excluded from certain folds, leading to biased performance estimates.

- **SMOTE Oversampling**:
  - Addressing Class Imbalance: SMOTE is employed to alleviate class imbalance by oversampling the minority class samples in the training data. This technique generates synthetic samples for the minority class, thereby improving its representation in the dataset and reducing the risk of model bias towards the majority class.
  - Application During Training Fold Only: SMOTE is applied exclusively during the training fold to prevent synthetic samples from contaminating the testing and validation folds. This ensures that the model's performance is evaluated on genuinely unseen data, reflecting its ability to generalize to real-world scenarios.
- **Model Training and Evaluation**:
  - Training Procedure: The model is trained using the resampled training data, incorporating SMOTE-generated samples to enhance its ability to learn from minority class instances effectively.
  - Evaluation on Testing Fold: After training, the model is evaluated on the corresponding testing fold to assess its generalization performance. By testing on unseen data, we obtain a more accurate estimation of the model's ability to classify land cover types accurately.
- **Result Aggregation**:
  - Performance Metrics Collection: The accuracy results from each fold of every repetition are aggregated to compute summary statistics, such as mean and standard deviation. This comprehensive analysis provides insights into the model's overall performance and its variability across different subsets of the data.
- This validation strategy ensures robust evaluation of the model's performance under varying conditions, addressing class imbalance through SMOTE oversampling while maintaining the integrity of the cross-validation procedure.
- By incorporating both Stratified K-Fold cross-validation and SMOTE oversampling, the model's performance estimation becomes more reliable and generalizable, contributing to the credibility and effectiveness of the land cover classification system.

3.3.9 Results

The performance of the proposed land cover classification model was evaluated based on training and testing accuracy metrics, as well as the overall model training time.

- **Training Accuracy**: The model achieved an impressive training accuracy of 97.909%, demonstrating its ability to effectively learn from the training data and capture the underlying patterns associated with different land cover types.

```
accuracy = np.mean(results)

print("Accuracy:", accuracy)
```

```
Accuracy: 0.9790960550308228
```

Figure 3.12: Mean Training Accuracy

- **Testing Accuracy**: Upon evaluation on unseen testing data, the model exhibited strong generalization performance, achieving a testing accuracy of 97.32%. The final predicted map is depicted in Figure 3.14, and the corresponding Confusion Matrix is depicted in Figure 3.15.

```
from sklearn.metrics import accuracy_score

accuracy = accuracy_score(gt.reshape((21025)), y_preds)
print('Accuracy:', accuracy)
```

```
Accuracy: 0.9732223543400713
```

Figure 3.13: Test Accuracy

- **Model Training Time**: The entire model training process took approximately 1hr 22 minutes to complete. This includes data preprocessing, model compilation, training iterations, and evaluation steps. Despite the complexity of the model and the size of the dataset, the training time remains reasonable, indicating efficient utilization of computational resources.

These results underscore the effectiveness of the proposed land cover classification model in accurately predicting land cover types from hyperspectral imagery. The high training and testing accuracies, coupled with reasonable training time, validate the model's practical utility and its potential for real-world deployment.

3.3.10 Technologies Used

The various technologies used for land use and land cover classification with deep learning are as listed below:

- **Python**: Python is a versatile, high-level programming language known for its readability and simplicity. Widely used in various domains, including web development,

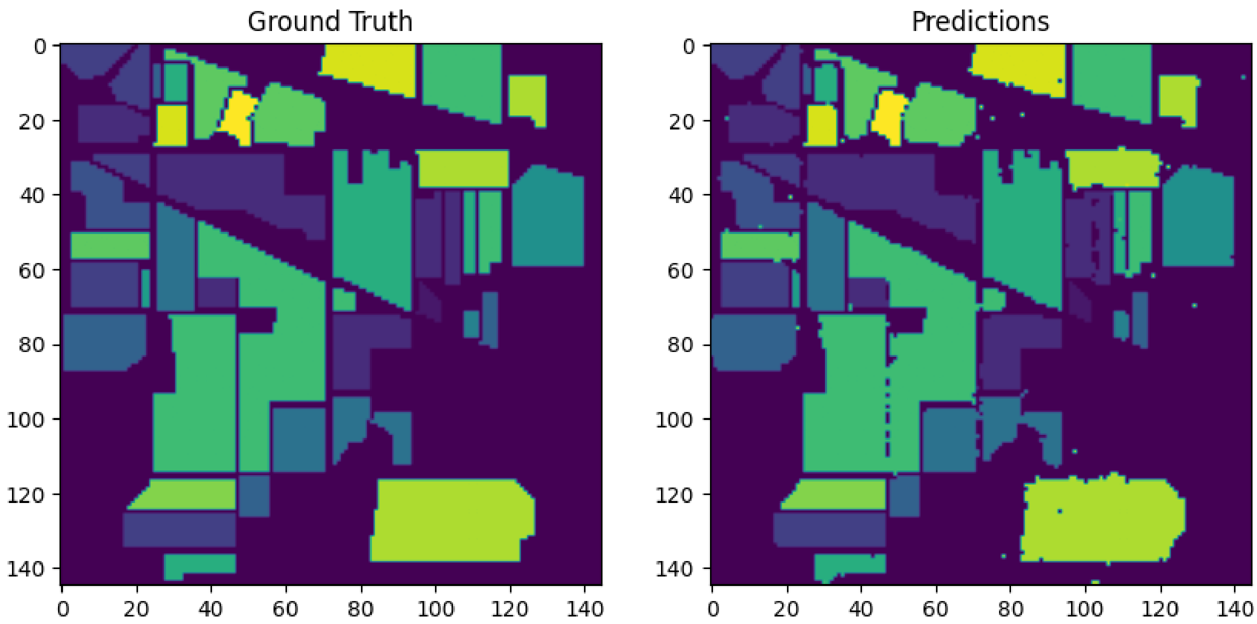

Figure 3.14: Ground Truth vs Model Prediction

data science, machine learning, and more, Python has a large and active community. Its extensive standard library and support for third-party packages make it a popular choice for diverse applications, from scripting to complex software development.

- **Pandas**: Pandas is a powerful Python library for data manipulation and analysis. It provides data structures like DataFrame and Series, making it efficient to work with structured data. Pandas simplifies tasks such as cleaning, exploring, and transforming data, offering a high-level interface for data manipulation.

- **NumPy**: NumPy is a fundamental library for numerical computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays. NumPy is essential for scientific computing tasks and forms the foundation for various other libraries in the Python ecosystem.

- **Scikit-learn**: Scikit-learn is a machine learning library for Python that offers simple and efficient tools for data analysis and modeling. It includes various algorithms for classification, regression, clustering, and dimensionality reduction. Scikit-learn is user-friendly, making it accessible for both beginners and experts in machine learning.

- **Imbalanced Learn**: Imbalanced Learn a Python library specifically designed for tackling class imbalance in machine learning datasets. One of the key techniques utilized from imblearn is Synthetic Minority Over-sampling Technique (SMOTE), which addresses class imbalance by generating synthetic samples for the minority class.

- **TensorFlow**: TensorFlow is an open-source machine learning framework by Google, which was instrumental in our project for its flexible architecture and high-level APIs. It facilitated model development, data processing, and deployment tasks efficiently.

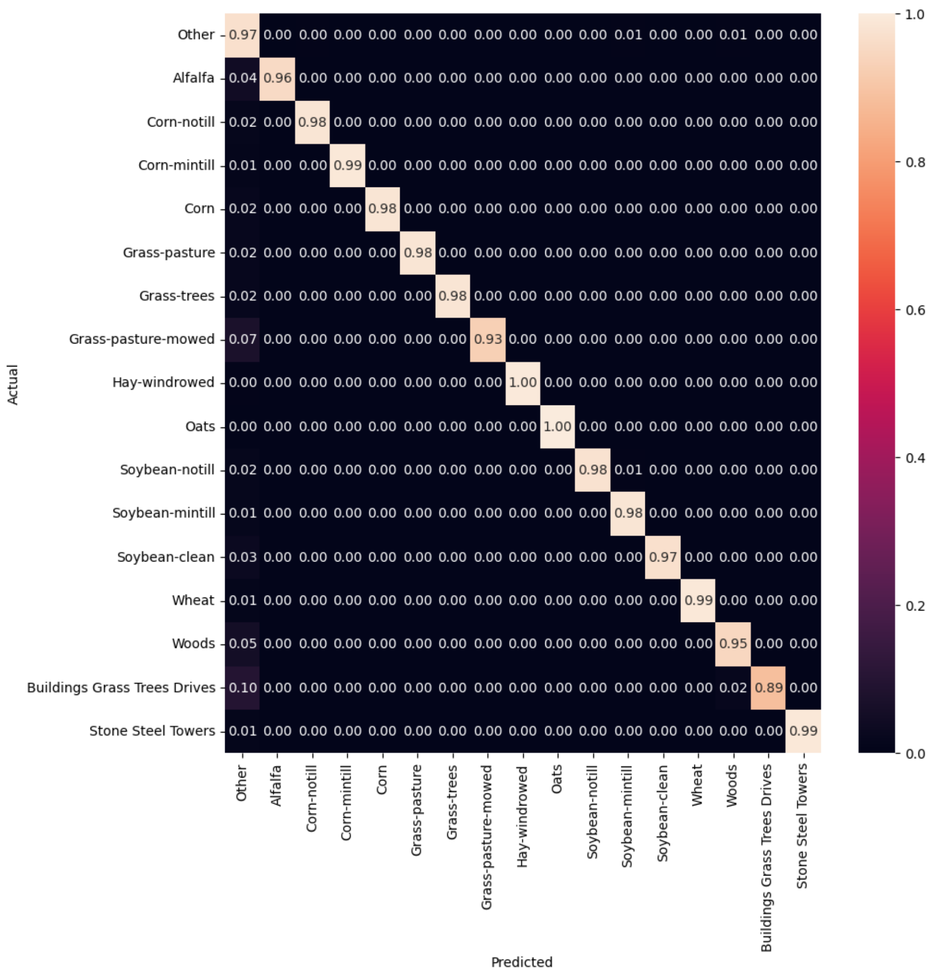

Figure 3.15: Confusion Matrix

TensorFlow's scalability and extensive library enriched our project with advanced machine learning capabilities, ensuring its success.

- **Google Colab**: Google Colab is a cloud-based Jupyter notebook environment, played a pivotal role in our project by providing free access to powerful GPUs and TPUs. Its seamless integration with Google Drive enabled collaborative work and easy sharing of notebooks. With built-in libraries and support for various languages, Colab accelerated our development process significantly.

Chapter 4

Conclusion and Future Works

4.1 Conclusion

In conclusion, our research project represents a significant step forward in the field of LULC classification using HSI data. By developing a simplified CNN model and integrating domain knowledge from Remote Sensing, specifically the NDVI, we were able to enhance the accuracy and reliability of our classification approach. Through careful consideration of class imbalance and the implementation of SMOTE alongside stratified k-fold cross-validation, we ensured that our model could effectively generalize to diverse land cover types.

The utilization of SMOTE not only addressed class imbalance but also improved the overall performance of the model by ensuring adequate representation of all classes, even those that were initially under-represented. Our findings underscore the importance of combining deep learning techniques with domain expertise to tackle complex challenges in HSI classification. By demonstrating the effectiveness of our methodology, we provide valuable insights for future research endeavors aimed at leveraging advanced technologies and domain knowledge to improve land cover classification accuracy and support informed decision-making in various domains, including urban planning, environmental management, and natural resource conservation.

4.1.1 Future Works

In our future endeavors, we plan to augment our methodology by integrating super-resolution techniques to address the inherent spatial resolution limitations of HSI data. By enhancing spatial detail, we aim to improve the precision and accuracy of land cover classification. This enhancement will enable us to capture fine-scale features and subtle variations in land cover more effectively, thereby refining classification results and providing more valuable insights into Earth's surface dynamics.

Furthermore, we intend to explore fusion methods that combine HSI data with other high-resolution spatial data sources such as LiDAR or multispectral imagery. By leveraging complementary information from multiple sources, we can enhance the discriminative power of our classification models and improve their overall performance. This approach holds the promise of producing more comprehensive and reliable land cover classification results, particularly in complex and heterogeneous landscapes.

Additionally, we see potential in leveraging deep learning techniques for feature extraction and representation learning in our classification framework. By harnessing the hierarchical features learned by deep neural networks, we can further enhance the efficiency and robustness of our classification methodology. This advancement not only improves classification accuracy but also streamlines the process, making it more adaptable to a wide range of applications in environmental management and resource conservation.

References

[1] Bhosle, K., Musande, V. Evaluation of Deep Learning CNN Model for Land Use Land Cover Classification and Crop Identification Using Hyperspectral Remote Sensing Images. J Indian Soc Remote Sens 47, 19491958 (2019). https://doi.org/10.1007/s12524-019-01041-2

[2] Carranza-García M, García-Gutiérrez J, Riquelme JC. A Framework for Evaluating Land Use and Land Cover Classification Using Convolutional Neural Networks. Remote Sensing. 2019; 11(3):274. https://doi.org/10.3390/rs11030274

[3] Rasheed, M.U., Mahmood, S.A. A framework based on deep neural network (DNN) for land use land cover (LULC) and rice crop classification without using survey data. Clim Dyn 61, 56295652 (2023). https://doi.org/10.1007/s00382-023-06874-9

[4] Vali A, Comai S, Matteucci M. Deep Learning for Land Use and Land Cover Classification Based on Hyperspectral and Multispectral Earth Observation Data: A Review. Remote Sensing. 2020; 12(15):2495. https://doi.org/10.3390/rs12152495

[5] H. Ghanbari, M. Mahdianpari, S. Homayouni and F. Mohammadimanesh, "A Meta-Analysis of Convolutional Neural Networks for Remote Sensing Applications," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 3602-3613, 2021, doi: 10.1109/JSTARS.2021.3065569

[6] Prasad S Thenkabail, Ronald B Smith, Eddy De Pauw, "Hyperspectral Vegetation Indices and Their Relationships with Agricultural Crop Characteristics", in Remote Sensing of Environment, vol. 71, Issue 2, 2000, pp. 158-182, doi: https://doi.org/10.1016/S0034-4257(99)00067-X.