**A Project Report**

*on*

Land Use and Land Cover (LULC) Classification Using Deep Learning

*carried out as part of the Deep Learning Lab* ***project***

*Submitted*

by

**<Arul Chudiwale>**

**<229309027>**

**Data Science and Engineering**

Under the Guidance of

**Guide Name**

**Ms. Shweta Redkar / Ms. Harshika Mathur**

**ABSTRACT**



The rapid pace of urban development and environmental degradation has increased the need for effective land use and land cover (LULC) classification methods.

This project utilizes a deep learning-based Convolutional Neural Network (CNN) to classify satellite images into various LULC categories, leveraging the EuroSAT dataset. The project further includes a time-series change detection feature that helps track and identify alterations in land cover over time, a significant improvement over manual image analysis. The model achieved over 92% training accuracy, demonstrating its capability as a robust tool for real-world applications such as urban planning, environmental monitoring, and disaster management.

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No** | **Table Title** | **Page No** |
|  |  |  |
| 1 | EuroSAT Dataset Composition |  |
| 2 | |  | | --- | | Model Performance Metrics |  |  | | --- | |  | |  |
|  |  |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No** | **Figure Title** | **Page No** |
|  |  |  |
| 1 | |  | | --- | | System Architecture |  |  |  | | --- | --- | |  |  | |  |
| 2 | |  | | --- | | Model Accuracy Plot |  |  | | --- | |  | |  |
| 3 | |  | | --- | | Training Metrics Plot |  |  | | --- | |  | |  |
| 4 | |  | | --- | | Change Detection Illustration |  |  | | --- | |  | |  |

**Table of Contents**

1. **Introduction**
   * **1.1 Overview and Motivation**
   * **1.2 Problem Statement**
   * **1.3 Objectives**
   * **1.4 Scope of the Project**
2. **Background**
   * **2.1 Conceptual Overview and Literature Review**
   * **2.2 Related Methodologies and Existing Works**
3. **System Design and Methodology**
   * **3.1 System Architecture**
   * **3.2 Development Environment**
   * **3.3 Methodology: Algorithm and Procedures**
4. **Implementation and Results**
   * **4.1 Modules and Classes of the Project**
   * **4.2 Implementation Details**
   * **4.3 Results and Analysis**
   * **4.4 Timeline Chart**
5. **Conclusion and Future Work**
6. **References**

**1. Introduction**

**1.1 Overview and Motivation**

With the growth of urban areas and ongoing environmental changes, LULC classification has become a pivotal tool for sustainable development. The motivation for this project lies in the need for automated and accurate solutions for classifying satellite imagery, which supports timely decision-making in urban planning, agricultural management, and environmental conservation. Deep learning, specifically Convolutional Neural Networks (CNNs), has proven to be a powerful method for image classification due to its capability to learn complex spatial hierarchies and patterns within data.

**1.2 Problem Statement**

Manual LULC classification methods are labor-intensive and prone to human error. This project addresses the challenge of developing an automated deep learning model to classify satellite images accurately and integrate a time-series feature to detect changes in land cover over time.

**1.3 Objectives**

* Develop a CNN-based model for classifying satellite images into predefined LULC categories.
* Build an interactive application for users to upload and classify satellite images in real time.
* Implement a change detection feature to highlight differences in LULC over time.
* Evaluate the model's performance using standard metrics and test it on real-world images.

**1.4 Scope of the Project**

The scope of this project includes training a deep learning model on the EuroSAT dataset, developing an interactive front-end application with Streamlit, and incorporating change detection capabilities for analyzing time-series satellite images.

**2. Background**

**2.1 Conceptual Overview and Literature Review**

Convolutional Neural Networks (CNNs) are widely recognized for their efficacy in image classification tasks due to their layered structure that captures spatial and hierarchical features in data. The EuroSAT dataset, derived from Sentinel-2 satellite images, is a benchmark for LULC classification and consists of ten classes such as forests, water bodies, and urban areas (Helber et al., 2019). Previous studies have shown that transfer learning with pre-trained models, like MobileNetV2, significantly enhances performance while reducing training time (Howard et al., 2017).

Change detection, which identifies alterations between images taken at different times, is a critical task in remote sensing. Traditional change detection relies on simple image differencing methods, whereas deep learning-based approaches offer greater accuracy and automation (Singh, 1989).

**2.2 Related Methodologies and Existing Works**

Studies have utilized various deep learning architectures to improve LULC classification. Methods such as data augmentation and transfer learning are often employed to enhance generalization. Existing applications are limited in their ability to offer interactive, real-time change detection, which this project addresses through an integrated Streamlit application.

**3. System Design and Methodology**

**3.1 System Architecture**

The project is built upon a modular system architecture that includes:

* **Data Preprocessing Module**: Normalizes and resizes input images.
* **CNN Model**: Composed of three convolutional layers, each followed by max-pooling layers, with a final dense output layer using softmax activation.
* **Front-End Application**: A Streamlit-based interface for image uploads and results display.
* **Change Detection Module**: Analyzes sequential images to detect land cover changes.

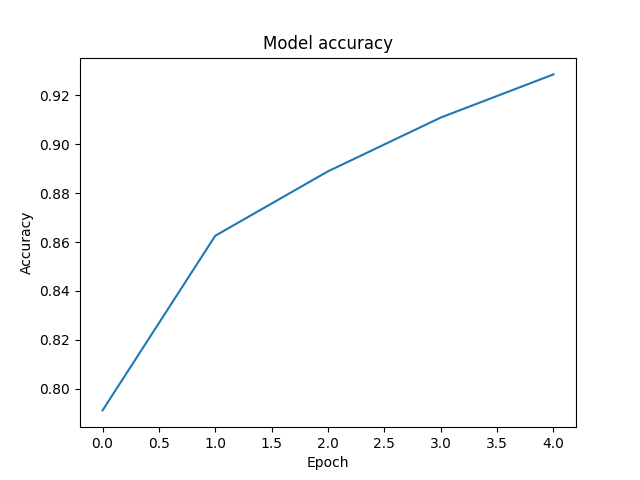
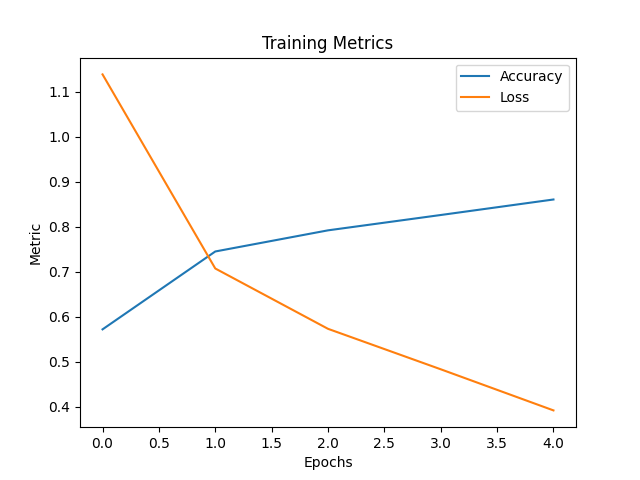
**Figure 1** illustrates the architecture of the system, showing the data flow from input to output.

**3.2 Development Environment**

* **Software**: Python 3.x, TensorFlow, Keras, Streamlit, and supporting libraries such as Pillow and Matplotlib.

**3.3 Methodology: Algorithm and Procedures**

1. **Data Preprocessing**:
   * Normalize pixel values to a [0, 1] range.
   * Resize images to 64x64 pixels for model compatibility.
2. **Model Development**:
   * The model comprises three convolutional layers(CNN), each followed by max-pooling.
   * A fully connected layer with 128 units and ReLU activation is used before the output layer.
   * The model was enhanced to include training metrics tracking, such as accuracy and loss, to evaluate its performance over the epochs.
3. **Training**:
   * The model was trained with sparse\_categorical\_crossentropy loss and the Adam optimizer over five epochs.
   * Training data was shuffled and batched to optimize learning.
   * The trained model's accuracy was calculated using the accuracy metric during training, which helped track the model’s performance across each epoch. The training process included plotting accuracy and loss metrics to visualize the learning curve and assess the model's progress.
4. **Application Implementation**:
   * Streamlit was used to develop a user-friendly web interface.
   * Users can upload single or multiple images for LULC classification and change detection.



**4. Implementation and Results**

**4.1 Modules and Classes of the Project**

The project consists of several modules:

* **DataLoader**: Loads and preprocesses the dataset.
* **ModelTrainer**: Defines and trains the CNN model.
* **AppHandler**: Manages the web-based interface.
* **ChangeDetector**: Compares and highlights changes between image classifications.

**4.2 Implementation Details**

The lulc\_classifier.py script handles model training and saves the trained model (lulc\_model.h5). This script also includes the plotting of training metrics to display accuracy and loss over epochs, providing an insight into the training process. The app.py file was updated to show overall accuracy and per-class accuracy metrics in the sidebar, offering users a comprehensive overview of the model’s performance during inference.

The addition of confidence scores for each prediction allows users to understand how certain the model is about its classifications. This enhancement, along with the accuracy metrics, supports better interpretation of results and user trust in the model's outputs.

**4.3 Results and Analysis**

The model achieved over 92% accuracy during training. **Figure 2** displays the accuracy plot, and **Figure 3** shows the training metrics over five epochs. The time-series change detection feature successfully identified differences in land cover, making it a valuable tool for monitoring land changes.

**Figure 2: Model Accuracy Plot**

**Figure 3: Training Metrics Plot**

The results indicate that the model is effective for LULC classification and can serve as a foundational tool for more extensive environmental monitoring applications.

**4.4 Timeline Chart**

| **Month** | **Task** |
| --- | --- |
| Month 1 | Data Collection and Preprocessing |
| Month 2 | Model Training and Evaluation |
| Month 3 | Application Development and Testing |

**5. Conclusion and Future Work**

The project successfully demonstrated the feasibility of using CNNs for automated LULC classification and integrating change detection. The application is a practical tool for urban planners and environmental scientists. Future work may involve using multispectral data for higher accuracy and incorporating GIS integration for advanced spatial analysis.

**6. References**

* Howard, A. G., et al. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv preprint arXiv:1704.04861*.
* Helber, P., Bischke, B., Dengel, A., & Borth, D. (2019). EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*.
* Singh, A. (1989). Digital change detection techniques using remotely-sensed data. *International Journal of Remote Sensing*.
* TensorFlow Documentation for CNN model development.
* Streamlit Documentation for creating web applications.