

## LoopVerse 2025 – Machine Learning Module

### Problem Statement: EuroSAT Land Cover Classification with Deep Learning

Name	Laiba Maqsood
Enrollment	03-134222-041
Program	BSCS
Semester	6

---

## 1. Introduction

This project focuses on building a deep learning-based classification model for the EuroSAT dataset, which contains satellite images representing different types of land cover. The goal is to accurately classify images into one of ten classes such as Annual Crop, Forest, Highway, River, and more. The approach involves data preprocessing, dataset reduction, image cleaning, augmentation, and building convolutional neural networks (CNNs) to achieve high classification accuracy.

---

## 2. Data Preparation

### 2.1 Dataset Extraction and Organization

The original EuroSAT dataset was provided as a ZIP file. The first step involved checking if the extracted dataset folder (DATA\_DIR) exists. If not, the ZIP file was extracted to this directory, ensuring the dataset was ready for processing.

### 2.2 Dataset Reduction

To improve training efficiency, a reduced dataset was created by randomly selecting 1000 images from each class within the original dataset. The reduced dataset preserved the original class folder structure and was saved to a new directory (reduced\_dir).

### 2.3 Cleaning Images

The reduced dataset was scanned for images with large black patches, specifically those with more than 95% black pixels, as well as any corrupted or unreadable images. Such images were removed to maintain dataset quality. No images met the 95% black pixel threshold, so no images were removed during cleaning.

### 2.4 Dataset Summary

A count of images per class was performed, showing a total of 10,000 images distributed evenly across the 10 classes. This balanced dataset was split into training (8,000 images) and validation (2,000 images) subsets.

---

## 3. Data Loading and Augmentation

Images from the reduced dataset were loaded using TensorFlow's `image_dataset_from_directory` utility with an 80-20 split for training and validation. Before feeding the images into the model:

- A Gaussian blur filter was applied to smooth the images.
- Pixel values were rescaled from the range 0-255 to 0-1 for better model convergence.
- Data pipelines were optimized with caching, shuffling, and prefetching to improve training performance.

Validation confirmed that scaling was correctly applied, with pixel values ranging from approximately 0.045 to 1.0.

---

## **4. Model Development**

### **4.1 Initial CNN Model**

The first CNN model was built with three convolutional layers, each followed by max-pooling layers, to extract spatial features from the images. The output of the convolutional layers was flattened and passed through a fully connected hidden layer with 512 neurons, followed by an output layer with 10 neurons corresponding to the classes.

- Input shape: 150x150 pixels, 3 color channels
- Activation function: ReLU for hidden layers, Softmax for output layer

This model was compiled using the Adam optimizer and sparse categorical cross-entropy loss, suitable for multi-class classification with integer labels.

### **4.2 Training Results**

Training the initial model for 10 epochs resulted in:

- Training accuracy: 92%
  - Validation accuracy: 77%
- 

## **5. Improved CNN Model**

To enhance performance, a second CNN model was built by adding two more convolutional and max-pooling layers (five convolutional layers in total). The architecture retained the same fully connected layers at the end.

The optimizer was changed from Adam to RMSprop with a learning rate of 0.001, and the loss function remained sparse categorical cross-entropy.

Model2 with 5 CNN layers and RMSprop optimizer achieved strong performance, reaching up to around 80% validation accuracy, slightly better than Adam in some epochs. However, early epochs showed

similar accuracy 77% for both optimizers, indicating comparable initial learning. Overall, RMSprop helped improve validation accuracy slightly but results were fairly close between the two.

---

## **6. Conclusion**

The project successfully implemented a deep learning pipeline for EuroSAT land cover classification, including:

- Efficient dataset preparation and cleaning
- Data augmentation with Gaussian blur and pixel rescaling
- Building and training CNN models with increasing complexity
- Achieving promising accuracy with room for further improvements