"LoopVerse 2025: EuroSAT Land Cover Classification,"

NAME: IRTAZA IFTIKHAR , HADIA IMRAN

TEAM NAME: METAXPERTS

REPORT

- **Data Cleaning Process and Visual Proof of Filters Applied:** we start by mounting Google Drive to access the dataset. It then extracts the data from a zip file, checks for corrupted images, and applies different image processing techniques (resizing, converting to RGB, and applying filters). You can see visual proof of the filters applied by comparing the "Before Cleaning" and "After Cleaning" images and their color histograms in the output of the code cell. We used four filters gaussian, median, bilateral and non local denoising , we selected gaussian as it was most suitable.

GAUSSIAN

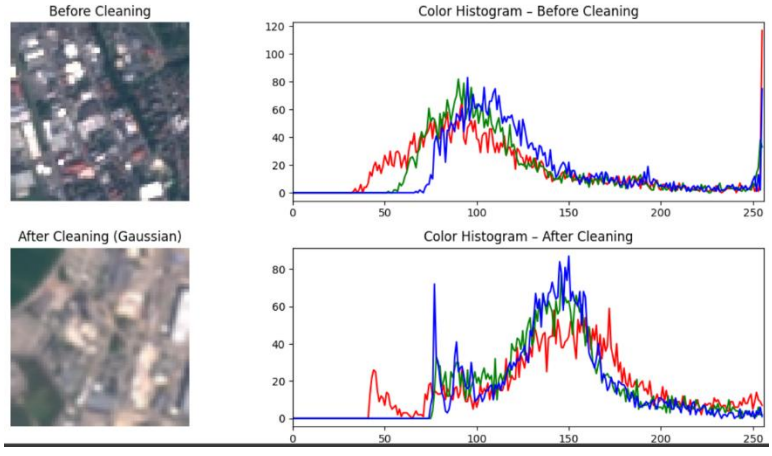
MEDIAN FILTER

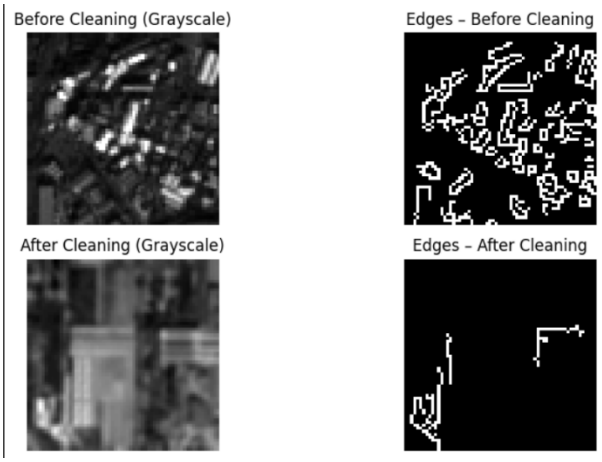
BILATERAL FILTER

NON LOCAL MEANS DENOISING

Here we selected **gaussian filter.**

- **Feature Visualization Plots:** we visualized random sample images from the "Industrial" class after different filtering steps in cells . This allows you to see the visual impact of the applied filters.





- **<u>CNN Architecture Diagram with Layer Details:</u>** The architecture of the CNN model is defined in the code cell. The model.summary() output in that cell provides a detailed breakdown of each layer, including the layer type (Conv2D, MaxPooling2D, Flatten, Dense, BatchNormalization, Dropout), the output shape of each layer, and the number of parameters in each layer. The activation functions used are primarily 'relu' for the hidden layers and 'softmax' for the output layer. The kernel sizes for the convolutional layers are (3,3), and the pool size for the MaxPooling layers is (2,2).

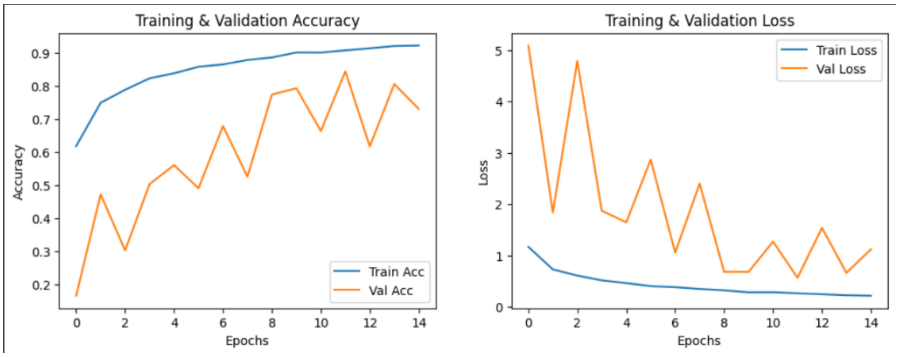
Model: "sequential_2"

Layer (type)	Output Shape	Param #
conv2d_6 (Conv2D)	(None, 62, 62, 32)	896
batch_normalization_8 (BatchNormalization)	(None, 62, 62, 32)	128
max_pooling2d_6 (MaxPooling2D)	(None, 31, 31, 32)	0
dropout_8 (Dropout)	(None, 31, 31, 32)	0
conv2d_7 (Conv2D)	(None, 29, 29, 64)	18,496
batch_normalization_9 (BatchNormalization)	(None, 29, 29, 64)	256
max_pooling2d_7 (MaxPooling2D)	(None, 14, 14, 64)	0
dropout_9 (Dropout)	(None, 14, 14, 64)	0
conv2d_8 (Conv2D)	(None, 12, 12, 128)	73,856
batch_normalization_10 (BatchNormalization)	(None, 12, 12, 128)	512
max_pooling2d_8 (MaxPooling2D)	(None, 6, 6, 128)	0
dropout_10 (Dropout)	(None, 6, 6, 128)	0
flatten_2 (Flatten)	(None, 4608)	0
dense_4 (Dense)	(None, 256)	1,179,904
batch_normalization_11 (BatchNormalization)	(None, 256)	1,024
dropout_11 (Dropout)	(None, 256)	0
dense_5 (Dense)	(None, 10)	2,570

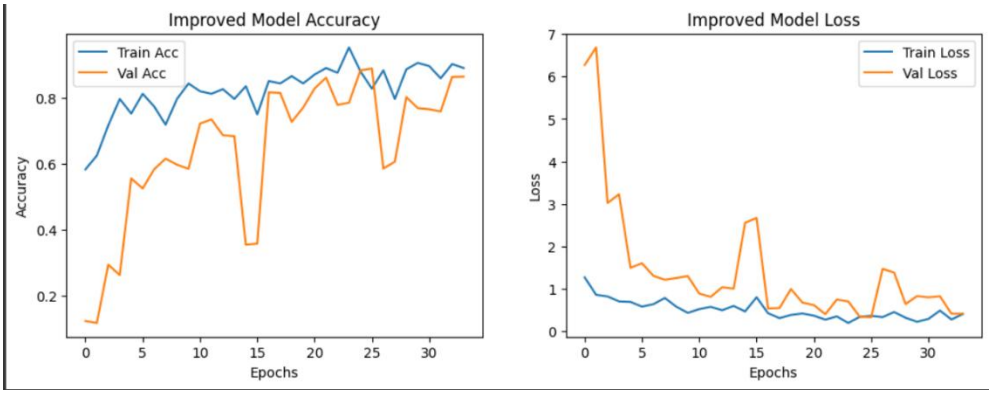
Total params: 1,277,642 (4.87 MB)
Trainable params: 1,276,682 (4.87 MB)
Non-trainable params: 960 (3.75 KB)

- **<u>Training and Validation Graphs Showing Accuracy and Loss Progression:</u>** The training process of the baseline CNN model is shown. The plots clearly show the progression of both training and validation accuracy and loss over the epochs. The improved model's training history is shown, including the effects of the learning rate scheduler and early stopping.

FOR BASELINE MODEL



FOR IMPROVED MODEL



- **Confusion Matrix and Metrics Table:** The evaluation of the improved model is performed. The output of this cell provides a classification report which includes precision, recall, f1-score, and support for each class, as well as overall accuracy, macro average, and weighted average. A confusion matrix is also displayed as a heatmap, visually representing the performance of the model for each class by showing the number of true positive, true negative, false positive, and false negative predictions.

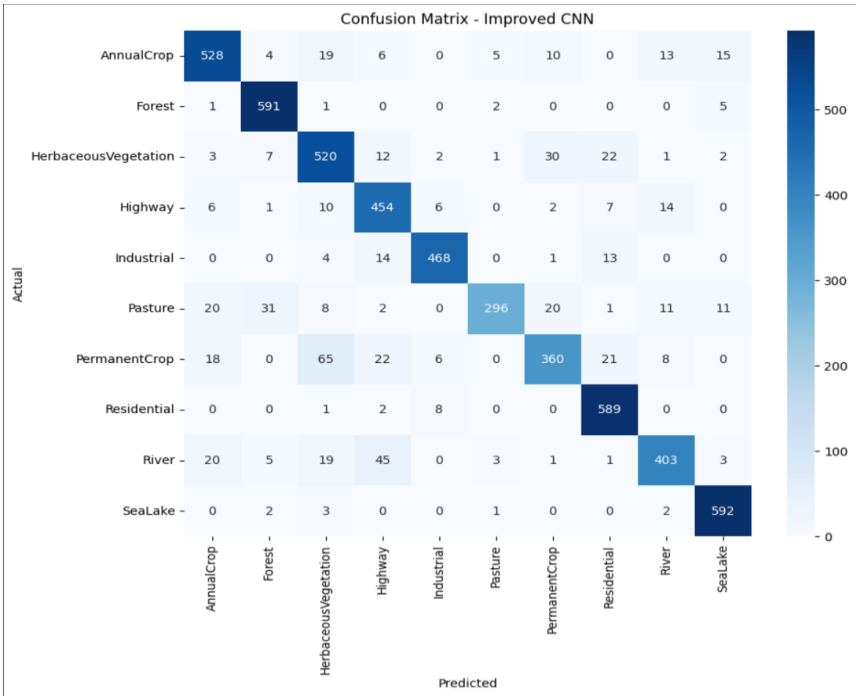
CLASSIFICATION REPORT

```
=== Classification Report ===
                     precision    recall  f1-score   support

         AnnualCrop       0.89      0.88      0.88       600
             Forest       0.92      0.98      0.95       600
HerbaceousVegetation       0.80      0.87      0.83       600
            Highway       0.82      0.91      0.86       500
         Industrial       0.96      0.94      0.95       500
            Pasture       0.96      0.74      0.84       400
      PermanentCrop       0.85      0.72      0.78       500
        Residential       0.90      0.98      0.94       600
              River       0.89      0.81      0.85       500
            SeaLake       0.94      0.99      0.96       600

           accuracy                           0.89      5400
          macro avg       0.89      0.88      0.88      5400
       weighted avg       0.89      0.89      0.89      5400
```

CONFUSION MATRIX



- **<u>Explanation of Algorithms and Techniques Used to Improve Accuracy:</u>** Several techniques were used to improve the accuracy of the model:
    - **Data Augmentation:** The ImageDataGenerator is used to apply random transformations to the training images (rotation, shifts, zoom, horizontal flip). This artificially increases the size and diversity of the training dataset, helping the model generalize better and reduce overfitting.
    - **Learning Rate Scheduler:** The ReduceLROnPlateau callback reduces the learning rate when the validation accuracy stops improving. This helps the optimizer make smaller steps in the loss landscape and potentially find a better minimum.
    - **Early Stopping:** The EarlyStopping callback monitors the validation accuracy and stops training if it doesn't improve for a certain number of epochs (patience). It also restores the model weights from the epoch with the best validation accuracy. This prevents overfitting and saves training time.
    - **Batch Normalization:** Batch normalization layers are added after the convolutional and dense layers in both models .This technique helps stabilize the training process and allows for higher learning rates.

- **Dropout:** Dropout layers are included after the pooling and dense layers in both models. Dropout randomly sets a fraction of the input units to zero during training, which helps prevent co-adaptation of neurons and reduces overfitting.
- **Improved CNN Architecture:** The improved model has a slightly different architecture with adjusted dropout rates compared to the baseline model .