EUROSAT LAND COVER CLASSIFICATION

INFO6147-Deep learning with python

Prepared by: Aiswarya Ayyante Valappil (1339261)

https://github.com/AiswaryaAV97/EUROSAT-LAND-COVER-CLASSIFICATION

Abstract

This project represents a deep learning-based approach to classify satellite images into various land cover types using the RGB version of the EuroSAT dataset. A Resnet-50 convolutional neural network, pretrained on imageNet, was fine-tuned on over 27,000RGB sentinel-s images of 10 classes such as Forest, Residential, River and Agricultural areas and so on. The model achieved high accuracy through data argumentation, normalization and optimized training. To enhance the interpretability, Grad-CAM was integrated to visualize class-specific regions influencing the model's predictions. Additionally, a user-friendly web interface was developed using streamlit and developed via ngrok, allowing real-time image upload and prediction. The system demonstrates the strong potential to show how the classification system can be used for detecting the land cover changes and how it can assist in improving geological maps.

Introduction

The main object of the project is to build a CNN- based classifier (ResNet-50) to accurately label the land covers from satellite images. The dataset contains over 27000 labeled satellite images of 10 land cover categories. ResNET-50 trained on imageNet has been used for this purpose, to recognise different land type such as forests, rivers, industrial areas and agricultural area. To improve the transparency of and explainability, GRAM-CAM has been used to generate visual heatmaps that show which part of an image the model focused on during its prediction. The final step is wrapped in a user-friendly streamlit web application, developed via ngrok, making it easy to upload an image and view predictions along with visual heatmaps.

Methodology

Dataset selection

Dataset source: https://zenodo.org/records/7711810#.ZAm3k-zMKEA

The dataset contains 27,000 sentinal-2 images of 10 classes. River, annual crop, pasture, forest, residential, sea lake, permanent crop, industrial, highway and Herbaceous Vegetation are included in the dataset. The RGB version of the dataset contains the R, G, B frequency bands encoded as JPEG images.

Data preprocessing

A series of data preprocessing steps has been done before training the model. First defined a preprocessing pipeline with the resize (225,224) to resize every image to 224*224 pixels which is the ResNet-50 needed input. And then converted the image to a PyTorch tensor and scales pixel values to [0,1] followed by the standard ImageNet normalization. Downloaded the EuroSAT RGB dataset and applies preprocessing pipeline to each image when it's loaded. The EuroSAT dataset then split into 80% training and 20% validation using random_split() to ensure the model is trained and evaluated on different data. Data loader has been loaded for training and validation data with a batch size of 32.

Model selection and architecture

ResNet-50 model pretrained on ImageNet using the latest IMAGENET1K_V2 weights has been used as the convolutional neural network to extract the spatial features from the images. ResNet-50 is a 50-layer deep variant of the ResNet architecture widely used in image classification, object detection and transfer learning tasks. The final fully connected layer (fc) replaced to output 10 classes, matching the EuroSAT land Cover categories.

Model Architecture overview (ResNet-50): This includes 1 initial conv layer and 16-layer bottleneck blocks (each with 3 conv layers) and 1 final fully connected layer. ReLU activation function and BatchNorm2d has been applied every convolution layer to introduce the non-linearity and stabilize training and reduce covariate shift.

Model Training

Training and evaluation have been done for fine-tuned ResNet-50 model on the EuroSAT dataset. Model is trained using CrossEntropyLoss and optimized via SGD with learning rate of 0.001 over 5 epochs, evaluated the performance using the validation accuracy and training loss and saved the best performance model based on validation accuracy. A separate prediction function allowed classifying individual satellite images, returning both the predicted land cover class and confidence score. Softmax function has been used to convert the logits (raw score) which are the output of the ResNet into a probability distribution across all the classes.

Evaluation

To evaluate the performance of the model, accuracy has been calculated. Here is the summary of the training progress over the 5 epochs.

Epoch 1:

- Loss: **1.8022**
- Training Accuracy: **50.9%**
- Validation Accuracy: **69.5%**

Epoch 2:

- Loss: **0.9513**
- Training Accuracy: **77.2%**
- Validation Accuracy: **84.3%**

Epoch 3:

- Loss: **0.5725**
- Training Accuracy: **85.2%**

- Validation Accuracy: **85.9%**

Epoch 4:

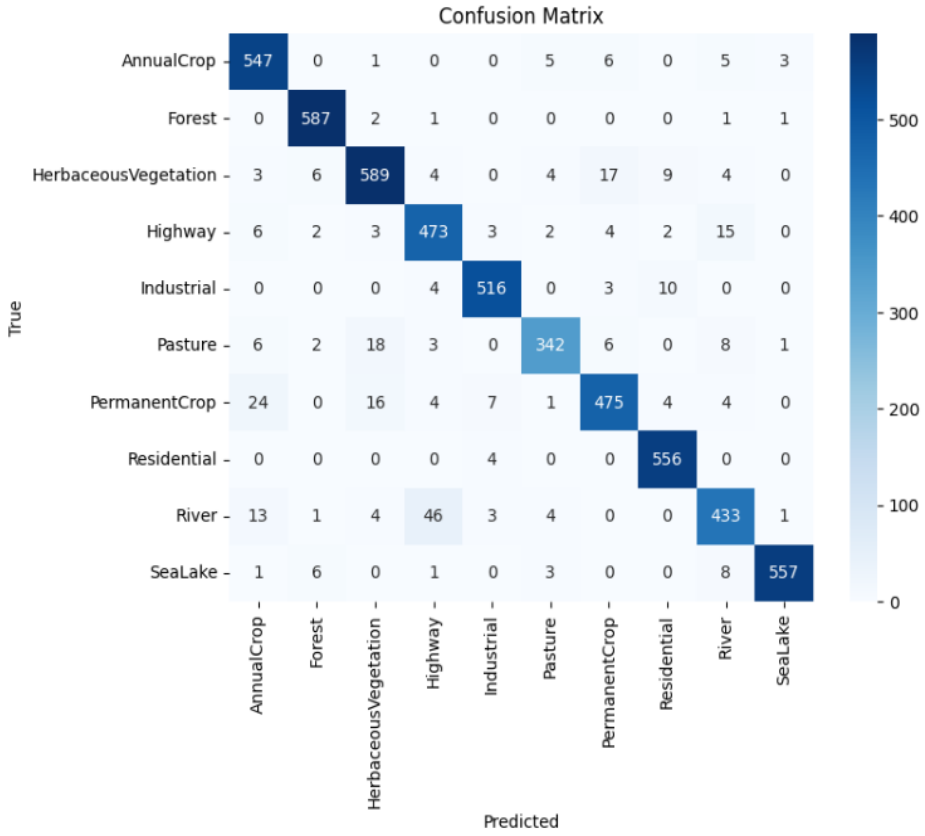
- Loss: **0.4003**
- Training Accuracy: **88.9%**
- Validation Accuracy: **92.0%**

Epoch 5:

- Loss: **0.3144**
- Training Accuracy: **90.7%**
- Validation Accuracy: **93.9%**

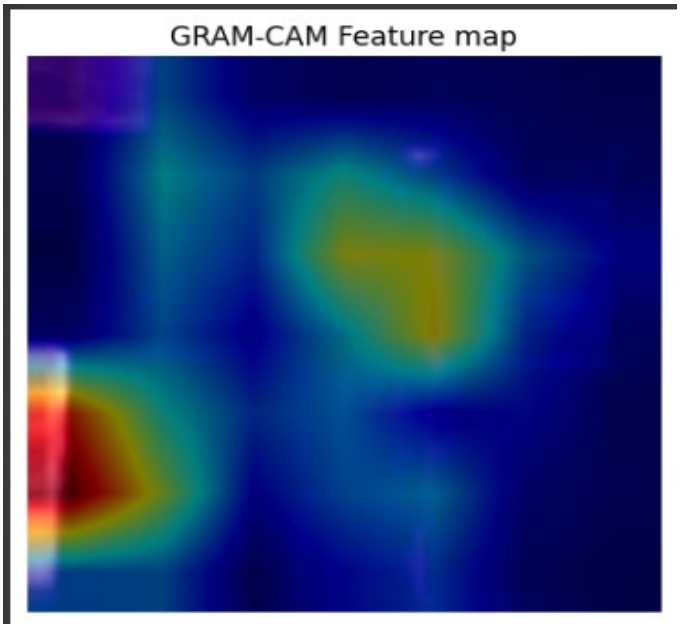
Summary:

The model shows steady improvement in both training and validation accuracy. The validation accuracy reached to 93.9% from 69.5% showing effective learning. And the best performed model saved at each epoch as accurately improved.



The figure shows the confusion matrix indicates that the ResNet-50 model performs exceptionally well in classifying most land cover types from the EuroSAT dataset, with high accuracy for categories like forest, herbaceous vegetation, sea lake and residential. However, the model struggles more with classes like river and pasture which are often misclassified due to visual similarity with other categories such as sea lake, annual crop and herbaceous vegetation. Overall, the model demonstrated strong classification capability, with minor confusion mostly arising between visually overlapping land types in satellite imagery.

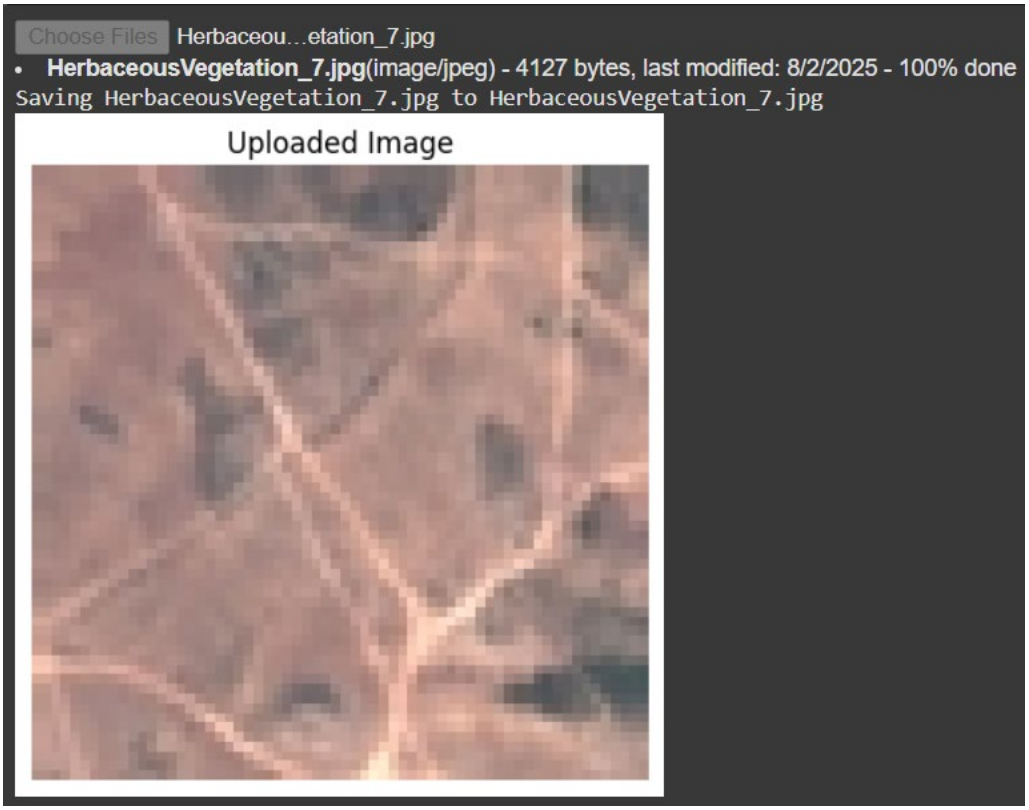
GRAD-CAM Feature map



This is a GRAD-CAM (Gradient-weighted class activation mapping) feature map of class0(annual crop), which highlights the important regions in the image that the CNN focused on while making the predictions. The Red/yellow areas represent regions with high influence on model's predictions and the blue areas indicate the low contribution to the classification. Also it enhances the model explainability by showing where the model is looking when classifying satellite images.

Results

To enable real-time predictions, an image upload feature was implemented. Using the upload menu, upload the satellite images directly. The uploaded image is read using python imaging library (PIL) and converted to RGB format to ensure compatibility with the trained model. It is then displayed using matplotlib.pyplot to provide immediate visual feedback to the user. After uploading the image, the predict_resnet() function is used to classify the land cover category. The image is first transformed using the same pipeline used for the training data, then passed through the fine-tuned ResNet-50 model. The model outputs a set of probabilities for each of the ten EuroSAT land cover classes. Then the class with highest probabilities is selected as the final prediction, a SoftMax function is applied to compute the model's confidence score. The predicted label and its corresponding confidence are printed for user interpretation. The below figure shows the upload menu and the displayed uploaded image.



Integration with GPT-4o

To enhance the interpretability along with the ResNet-50 predictions, GPT-4o's multi-model capabilities were integrated into the project. A custom function converts the uploaded satellite image into a base64 format and send it to GPT-4o via API, along with the prompt asking the model to classify the image into one of the EuroSAT land cover categories. GPT-4o processed both the image and the text instruction and returns a detailed response with its prediction.

Streamlit web application integration

To make the classification model accessible and interactive, a web application was developed using streamlit. The app.py scripts serves the backend for this app. It allows users to upload satellite images, which are then preprocessed and classified in real time using a fine-tuned ResNet-50 model.

When user uploads an image, the app preprocesses it using standard ImageNet normalization and resize it to 224*224 pixels. The image is then passed through the ResNet-50 model, which outputs a predicted land cover class along with the confidence score. The GRD-CAM algorithm was integrated, generating the heatmap that visually highlights which part of the image influenced the model's predictions. The entire setup is wrapped inside a streamlit interface with user-friendly prompts, image upload functionality, prediction outputs and heatmap, all these accessible directly in a browser. The output of the UI looks like the below pictures.

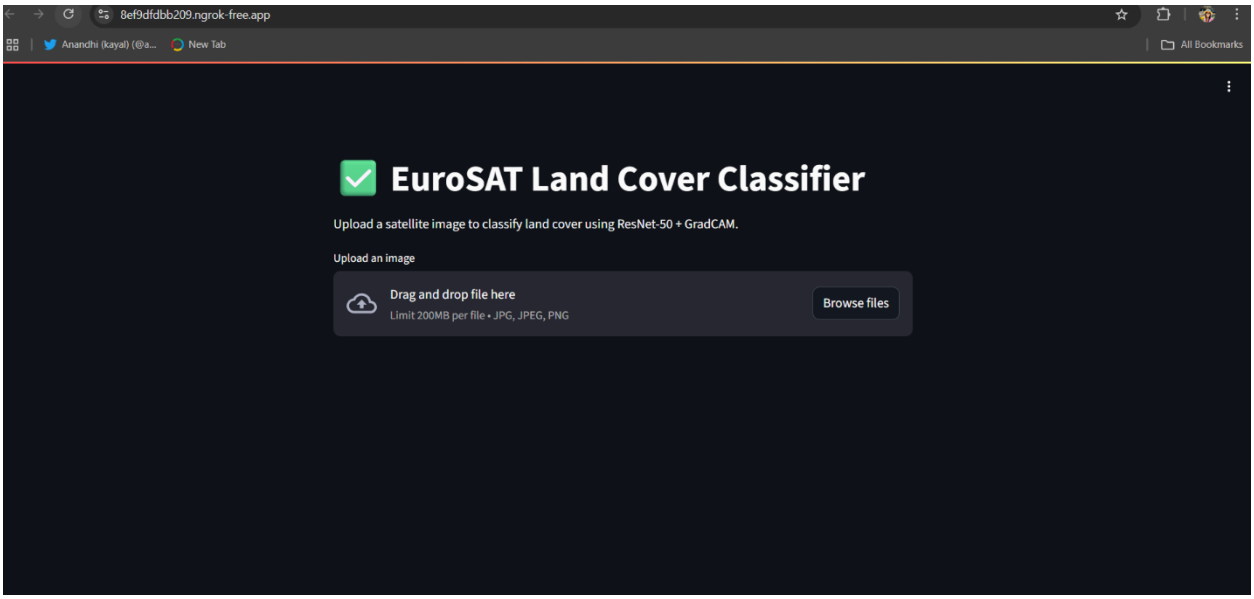


Figure:2 shows the UI using ngrok

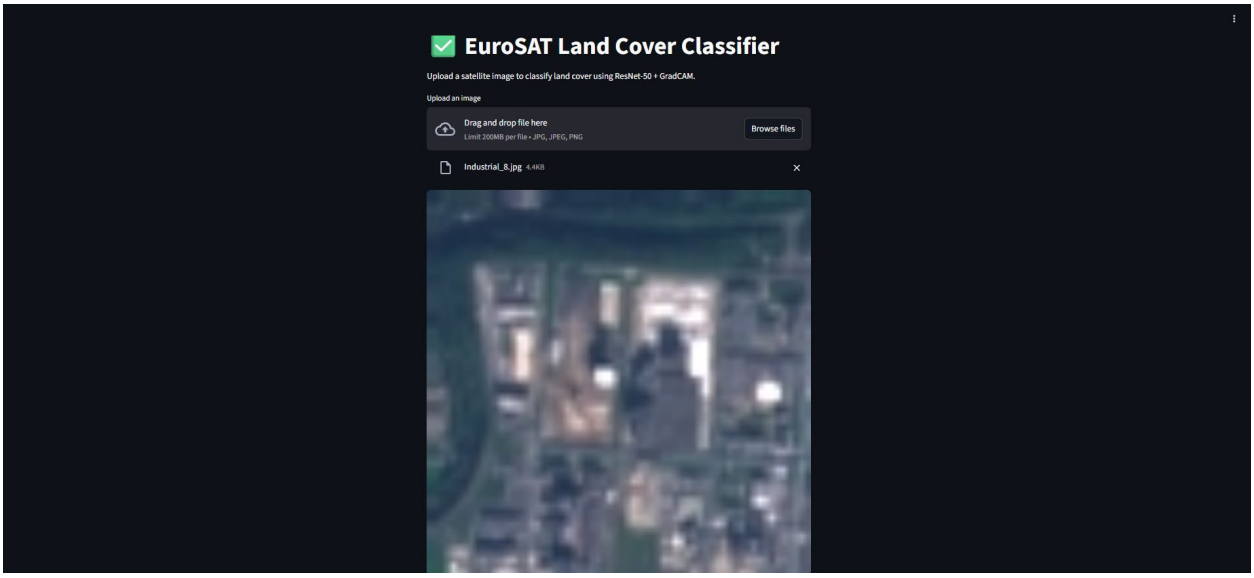


Figure:3 shows the Uploaded image via upload option and showing its prediction.

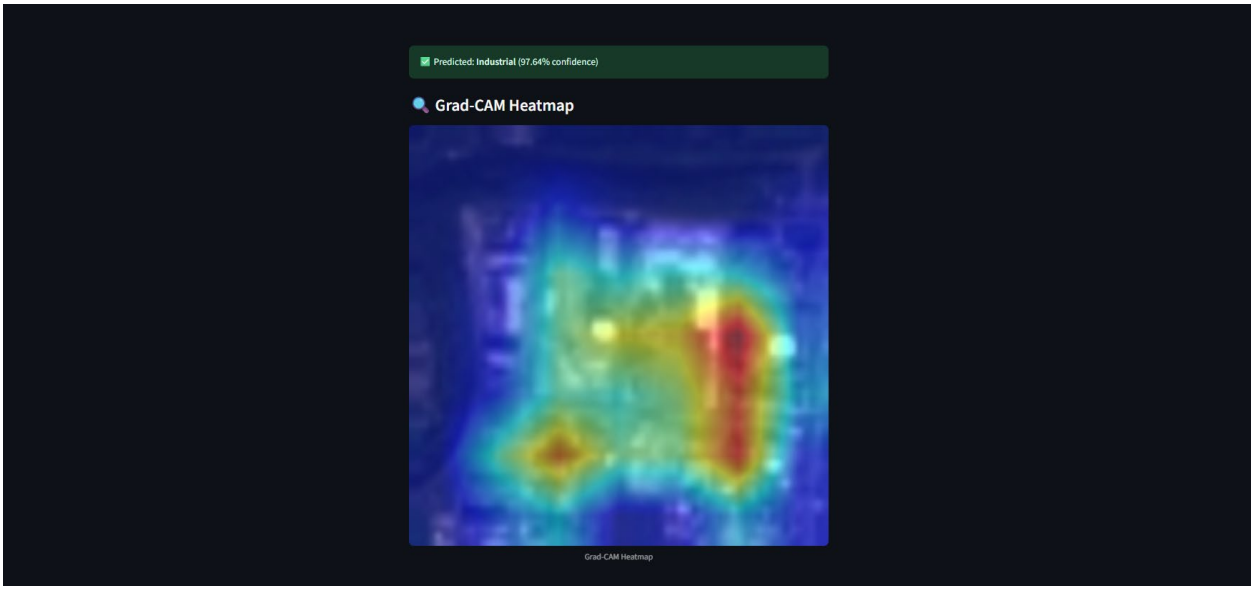


Figure4: Shows the heatmap of the image

Conclusion

This project successfully demonstrated the power of deep learning, specifically the ResNet-50, in automating land cover classification using satellite imagery. By fine-tuning a pretrained ResNet-50 model on the EuroSAT datasets, achieved high classification accuracy, with validation accuracy of 93% by the final epoch. The model effectively distinguished between 10 different land cover types, proving its capability in handling real-world remote sensing data. To enhance the interpretability, GRAD-CAM was integrated to visualize the heatmap of the images. The heatmaps help users understand which part of the image influenced the model's prediction.

The entire system was deployed in a streamlit web UI, allowing users to upload their own satellite images, receive the predictions and view the GRAD-CAM images from a browser. In summary, this project illustrates a full pipeline from model training to deployment, highlighting how deep learning can transform a satellite image into a readable geographic insight.

References

https://zenodo.org/records/7711810#.ZAm3k-zMKEA

file:///C:/Users/Jithin/Downloads/eurosat_paper.pdf

https://docs.pytorch.org/docs/stable/tensors.html

https://docs.pytorch.org/docs/stable/nn.html#normalization-layers

https://docs.pytorch.org

https://platform.openai.com

https://docs.streamlit.io/get-started/tutorials/create-an-app

https://arxiv.org/pdf/1610.02391

https://docs.kanaries.net/topics/Streamlit/streamlit-vs-dash