Earth Observation: Satellite Image Classification

BY

BENSON BABY THOMAS

crbenson7@gmail.com

07553358087

TABLE OF CONTENTS.

1.INTRODUCTION.

In the era of advanced technology and data science, the ability to extract meaningful insights from satellite imagery has revolutionized our understanding of the Earth's surface. Satellite images provide a wealth of information across diverse landscapes and environments, enabling applications ranging from urban planning and environmental monitoring to disaster response and resource management. This project, titled **"Earth Observation: Satellite Image Classification "**, aims to harness the power of machine learning to classify and analyse satellite imagery effectively.

2.DATASET.

The Geoeye-70 dataset is a meticulously curated collection of high-resolution Earth observation imagery acquired from a range of satellites. Established in April 2024, this dataset was designed to serve as a robust foundation for testing and refining pre-trained models, particularly in the realm of Vision Transformer technology.

With its detailed and diverse imagery, the Geoeye-70 dataset excels at capturing the intricate features of Earth's surface. It includes a wide variety of landscapes and cityscapes, making it ideal for comprehensive analysis and classification tasks. The dataset encompasses numerous classes, each representing different geographical and man-made features, thus providing a rich resource for developing and evaluating advanced image classification models.

In this project, we focused on a subset of 12 specific classes from the Geoeye-70 dataset: airport, beach, bridge, buildings, farmland, forest, harbour port, highway, island, lake, mountain, and river. This curated selection supports targeted classification tasks, facilitating more precise and relevant analysis of satellite imagery.

3.METHEDOLOGY.

3.1 Data Preparation and Preprocessing.

The dataset was organized into a directory structure with subdirectories representing specific class labels. File paths and labels were extracted to create lists for processing. The dataset was initially divided into training+validation and testing sets, with 15% of the data reserved for testing. The training+validation set was then further split, allocating approximately 15% for validation to ensure a balanced distribution across all phases. To enhance model generalization and robustness, an ImageDataGenerator was configured for the training set with augmentation techniques such as rotation, shifts, shearing, zooming, and horizontal flipping, while applying the preprocess_input function from MobileNetV2 for normalization. For validation and testing, separate ImageDataGenerator instances were used, applying only the preprocess_input function to maintain consistency and ensure accurate evaluation. The data was efficiently streamed from dataframes using ImageDataGenerator, facilitating smooth batch processing for training, validation, and testing.

3.2 Model Selection.

3.2.1 MobileNetV2.

For the classification task, MobileNetV2 was chosen due to its efficiency and suitability for edge devices. MobileNetV2 is a lightweight convolutional neural network known for its high performance and low computational cost, making it ideal for environments with limited resources. The model was initialized with pre-trained weights from the ImageNet dataset and configured without the top classification layers, allowing us to build a custom classifier. The architecture was adapted with a custom head comprising a GlobalAveragePooling2D layer, a Dense layer with 256 units and ReLU activation, a Dropout layer with a 0.5 rate to prevent overfitting, and a final Dense layer with softmax activation for multi-class classification. The model was compiled using the Adam optimizer and categorical crossentropy loss function. Training was conducted over a maximum of 100 epochs with early stopping configured to monitor validation loss. Early stopping was set with a patience of 20 epochs, which triggered at epoch 80, preventing overfitting and ensuring that the best model was saved based on validation performance.

3.3 Model Conversion.

To deploy the trained MobileNetV2 model on edge devices, it was converted to TensorFlow Lite (TFLite) format. The conversion process began by loading the trained Keras model using tf.keras.models.load_model. The TensorFlow Lite Converter was then utilized to transform the model into a TFLite-compatible format. Optimization flags were set to enhance the model's efficiency, including default optimizations for size and latency. Additionally, support for both TensorFlow Lite and TensorFlow operations was enabled to ensure broad compatibility. For further optimization, float16 quantization was applied, reducing the model size while maintaining a balance between performance and accuracy. The converted TFLite model was saved to a file named MobileNetv2Earth.tflite, completing the process. This conversion allows the model to run efficiently on edge devices, supporting real-time image classification in mobile and embedded environments.

4.EVALUATION AND TESTING.

4.1 Evaluation Metrics.

The performance of the MobileNetV2 model was assessed using several key metrics:

Accuracy: This metric measures the proportion of correctly classified images out of the total number of images. It provides an overall indication of the model's ability to correctly categorize images across all classes.

Precision: Precision evaluates the model's ability to correctly identify positive instances of a specific class out of all instances predicted as that class. It is defined as the ratio of true positives to the sum of true positives and false positives.

Recall: Recall, or sensitivity, measures the model's ability to identify all relevant instances of a specific class. It is defined as the ratio of true positives to the sum of true positives and false negatives.

F1 Score: The F1 Score is the harmonic mean of precision and recall, providing a single metric that balances both false positives and false negatives. It is particularly useful when the class distribution is imbalanced.

These metrics collectively provide a comprehensive evaluation of the model's performance, offering insights into its accuracy and reliability in classifying satellite images across the predefined categories.

4.2 Classification Report.

The classification report provides a detailed evaluation of the MobileNetV2 model's performance across the 12 predefined classes. This report includes key metrics such as precision, recall, and F1 score for each class, along with the support, which is the number of true instances for each class in the test set.

4.2.1 Classification Report of MobileNetV2

Classification Report:

	precision	recall	f1-score	support
Highway	0.99	1.00	0.99	80
airport	0.97	0.98	0.97	96
beach	1.00	0.97	0.99	69
bridge	0.94	0.97	0.96	88
buildings	1.00	1.00	1.00	85
farmland	0.98	0.93	0.96	70
forest	0.97	0.92	0.94	62
harbor_port	0.97	0.99	0.98	71
island	0.96	0.98	0.97	66
lake	0.99	0.97	0.98	80
mountain	0.94	0.97	0.95	64
river	0.96	0.97	0.96	69
accuracy			0.97	900
macro avg	0.97	0.97	0.97	900
weighted avg	0.97	0.97	0.97	900

Figure 1: Classification Report of MobileNetV2

4.2.2 Classification Report of MobileNetV2 Lite

Classification Report:

	precision	recall	f1-score	support
Highway	0.99	1.00	0.99	80
airport	0.97	0.98	0.97	96
beach	1.00	0.97	0.99	69
bridge	0.94	0.97	0.96	88
buildings	1.00	1.00	1.00	85
farmland	0.98	0.93	0.96	70
forest	0.97	0.92	0.94	62
harbor_port	0.97	0.99	0.98	71
island	0.96	0.98	0.97	66
lake	0.97	0.97	0.97	80
mountain	0.94	0.95	0.95	64
river	0.96	0.97	0.96	69
accuracy			0.97	900
macro avg	0.97	0.97	0.97	900
weighted avg	0.97	0.97	0.97	900

Figure 2: Classification Report of MobileNetV2 Lite

4.3 **Confusion Matrix.**

The confusion matrix offers an in-depth view of the classification model's performance by presenting a table that compares the predicted class labels against the true class labels for all instances in the dataset. This matrix is instrumental in evaluating the model's ability to correctly classify each category, revealing not only the number of correct predictions but also the types of errors made. Diagonal elements of the confusion matrix represent the number of correctly classified samples for each class, indicating how well the model performs for that specific category. Off-diagonal elements show the misclassifications, where the model's predictions deviate from the actual labels, highlighting the classes where the model confuses or misclassifies samples. Analyzing these values provides valuable insights into the model's strengths and weaknesses, helping to identify specific areas where the model excels and where it may require further tuning or additional training data. Overall, the confusion matrix serves as a crucial tool for understanding the model's detailed performance, enabling targeted improvements and refining the classification system to achieve more accurate and reliable predictions across different classes.

4.3.1 Confusion Metrics of MobileNetV2.

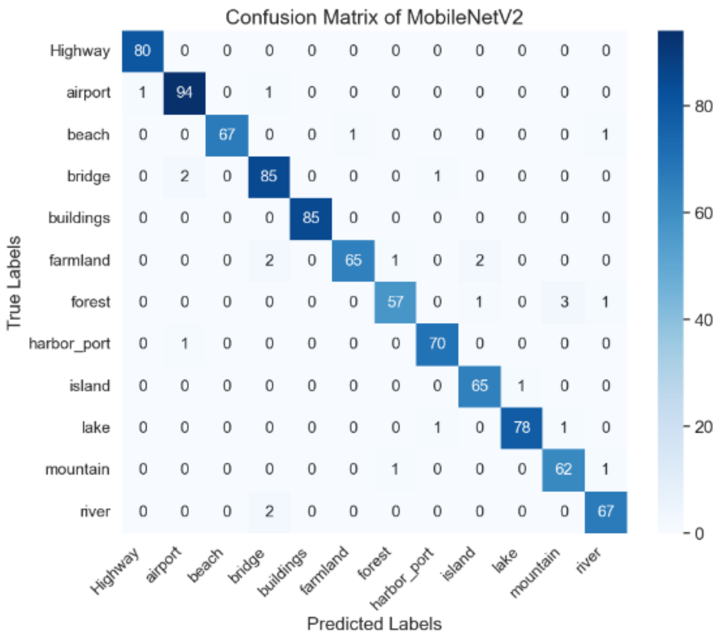

Figure 3: Confusion Metrics of MobileNetV2

4.3.2 Confusion Metrics of MobileNetV2 Lite.

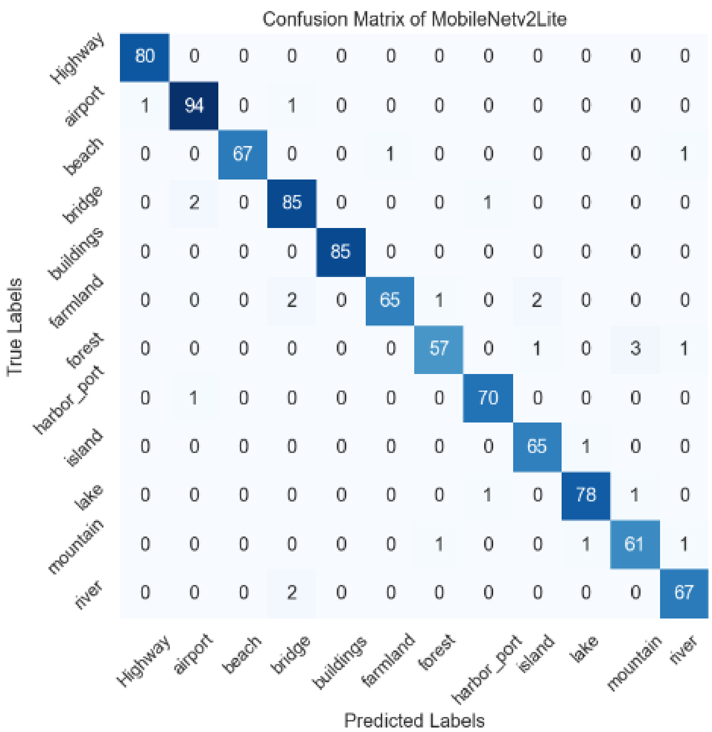

Figure 4: Confusion Metrics of MobileNetV2 Lite

4.4 Inference Time.

Inference time measures the duration required for the model to process a single image and generate a prediction. It is a critical factor in evaluating the efficiency and responsiveness of the model, particularly for real-time applications on edge devices. The average inference time for the TensorFlow Lite (TFLite) model was 0.013217 seconds, demonstrating its high efficiency and suitability for deployment on edge devices where quick processing is essential. In comparison, the average inference time for the Keras model was 0.072025 seconds. The significant reduction in inference time with the TFLite model underscores the benefits of model optimization for achieving faster processing and enhanced performance in edge device environments.

4.5 Model Evaluation Summary.

This section provides a comprehensive overview of the performance of the MobileNetV2 and MobileNetV2 Lite models based on key metrics such as test loss, accuracy, and average precision, recall, and F1 score. It also highlights the efficiency of each model in terms of inference time, offering insights into their effectiveness and suitability for various applications.

Model	Test Loss	Test Accuracy	Avg Precision	Avg Recall	Avg F1 Score	Inference Time	Total Test Samples
MobileNetV2	0.1284	0.9722	0.97	0.97	0.97	0.072025 seconds	900
MobileNetV2 Lite	0.1281	0.9711	0.97	0.97	0.97	0.013217 seconds	900

Table 1: Model Evaluation Summary

The Model Evaluation Summary table offers a comparative analysis of the standard MobileNetV2 and its optimized TensorFlow Lite version, focusing on key performance metrics. Both models exhibit high test accuracy, with MobileNetV2 achieving 97.22% and MobileNetV2 Lite slightly lower at 97.11%, alongside minimal test losses of 0.1284 and 0.1281, respectively. They both maintain an average precision, recall, and F1 score of 97%, demonstrating strong and consistent performance across all classes. The MobileNetV2 Lite model excels in efficiency with a significantly reduced inference time of 0.013217 seconds compared to 0.072025 seconds for the standard model, highlighting its suitability for deployment on edge devices where quick response times are essential.

5. CONCLUSION AND FUTURE SCOPE.

In this project, we utilized the MobileNetV2 model to classify satellite imagery into 12 distinct categories, including Highway, Airport, Beach, Bridge, Buildings, Farmland, Forest, Harbor Port, Island, Lake, Mountain, and River. The model achieved high accuracy and consistency across precision, recall, and F1 score, showcasing its effectiveness in distinguishing between these diverse land and infrastructure types. The TensorFlow Lite optimization further enhanced the model's efficiency, reducing inference time significantly and demonstrating its suitability for deployment on edge devices.

Looking ahead, there is considerable potential to extend this work by expanding the classification to include all 69 classes in the GeoEye-70 dataset. Such an enhancement would require fine-tuning the model to maintain high accuracy across a broader range of categories. Additionally, deploying the TensorFlow Lite model on small satellites could facilitate real-time classification, offering valuable insights directly from space. Another promising direction involves developing models to detect changes in structures or land over time, which could be instrumental in identifying and assessing the impact of natural disasters. This capability would enhance the monitoring and management of environmental and infrastructural changes, contributing to more effective disaster response and management strategies.

REFERENCE.

[1]. Mehmood, Maryam, Shahzad, Ahsan, Zafar, Bushra, Shabbir, Amsa, Ali, Nouman, Remote Sensing Image Classification: A Comprehensive Review and Applications, Mathematical Problems in Engineering, 2022, 5880959, 24 pages, 2022. https://doi.org/10.1155/2022/5880959

[2]. P. Helber, B. Bischke, A. Dengel and D. Borth, "EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 12, no. 7, pp. 2217-2226, July 2019, doi: 10.1109/JSTARS.2019.2918242.

[3]. Z. H. Jarrallah and M. A. A. Khodher, "Satellite Images Classification Using CNN :A Survey," 2022 International Conference on Data Science and Intelligent Computing (ICDSIC), Karbala, Iraq, 2022, pp. 111-116, doi: 10.1109/ICDSIC56987.2022.10075828.

[4]. H. Ouchra, A. Belangour and A. Erraissi, "Machine Learning Algorithms for Satellite Image Classification Using Google Earth Engine and Landsat Satellite Data: Morocco Case Study," in IEEE Access, vol. 11, pp. 71127-71142, 2023, doi: 10.1109/ACCESS.2023.3293828.