Group: 29

MobileDet for Object Detection in Remote Sensing Images

Members:
Lưu Bá Anh
Tạ Việt Đức
Phạm Ngọc Huy

1. Abstract

The abstract will provide a brief overview of the research objective, methods, and key findings. It should summarize the role of MobileDet in object detection when applied to remote sensing images, highlighting its efficiency and potential advantages for real-world applications.

2. Introduction

In the modern era of technological advancements, remote sensing has become a vital tool for environmental monitoring, urban planning, agricultural management, and disaster response. Remote sensing images captured by satellites or aerial platforms provide detailed insights into Earth's surface, enabling us to classify and detect various geographical features like forests, agriculture, and urban areas.

However, analyzing these massive volumes of data is complex and requires advanced computational methods. Traditional approaches rely on handcrafted features and are limited in terms of scalability and accuracy. With the advent of deep learning, specifically convolutional neural networks (CNNs), significant improvements in image classification and object detection have been achieved. This report explores how **MobileNetV2**, a lightweight deep learning model, can be effectively applied for object detection in remote sensing images.

3. Problem Statement

The primary problem being addressed in this project is the **automatic detection and classification of geographical features** in remote sensing images. Given a dataset of images representing various geographical categories like forests, river , and industrial areas, the goal is to build a model that can accurately classify these images into their respective categories

1. Varying scales and resolutions of satellite images.
2. Diverse geographical features that might look visually similar but differ in structure.
3. Limited availability of labeled data for training deep learning models.

4. Dataset Used

To evaluate the effectiveness of MobileNet for object detection, we used the **EuroSAT** dataset
The dataset contains a variety of object categories, including:

- AnnualCrop
- Forest
- HerbaceousVegetation
- Highway
- Industrial
- Pasture
- PermanentCrop
- Residential
- River
- SeaLake

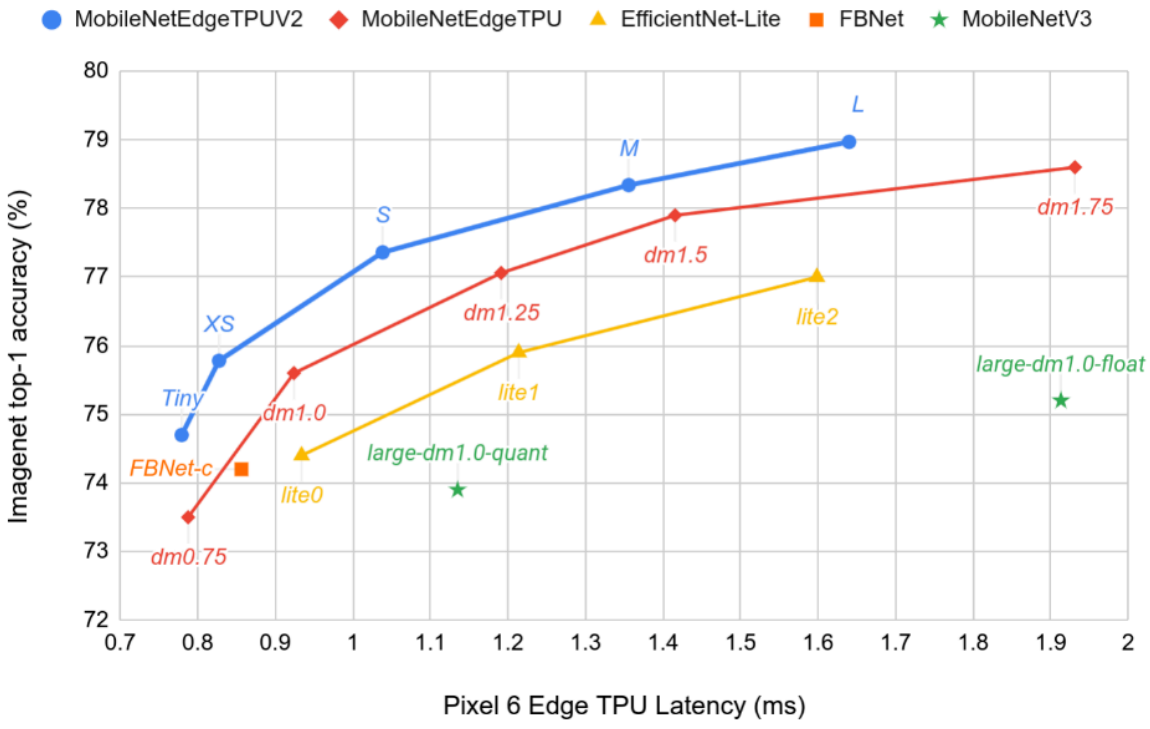
The dataset is organized into **training** and **validation** directories, with each directory containing subfolders for each class. The images in each class are of different geographical areas and conditions, providing the model with a wide variety of visual inputs to learn from.

5. Deep Learning Model: MobileNetV2

MobileNetV2 is a lightweight, efficient deep learning model that balances accuracy and speed. It was designed specifically for mobile and edge devices, making it suitable for remote sensing tasks that require real-time or near real-time predictions. MobileNetV2 employs depth wise separable convolutions, significantly reducing the computational complexity without sacrificing performance

Key features of MobileNetV2:

1. **Efficient Architecture**: Uses depthwise separable convolutions to reduce parameters and computation.
2. **Pre-trained on ImageNet**: Comes pre-trained on the large-scale ImageNet dataset, which helps transfer learning when applied to remote sensing data.
3. **Low Computational Cost**: Suitable for applications with resource constraints such as edge computing or mobile deployment.



6. Training Process

6.1. Data Augmentation (only available on hardcore training)

To improve the model's generalization ability, the training data is augmented using the **ImageDataGenerator** class in TensorFlow. Various transformations such as random rotations, shifts, shear, zoom, and flips are applied to each image in real-time during training.

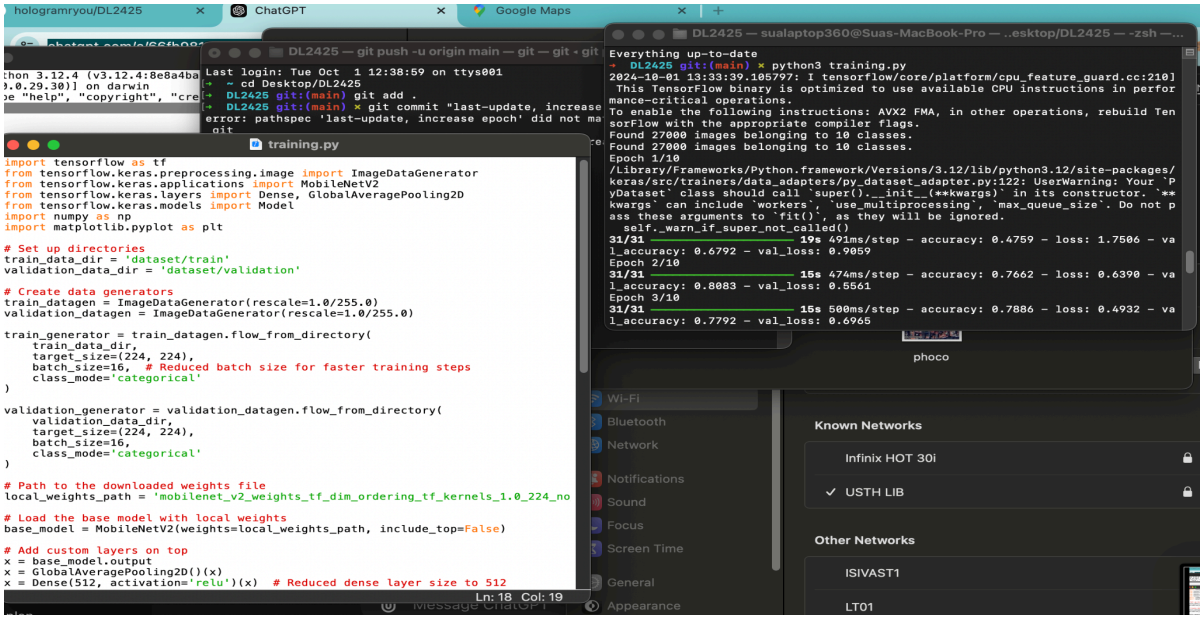
6.2. Training and Validation

The model is trained for 10 epochs with a batch size of 16. During training, both the training and validation accuracies are monitored. The data generators automatically feed images into the model in batches and apply augmentations in real-time.

$$\text{The total steps} = \text{Total samples} / \text{batch size}$$

To reduce the time to process, we can:

- Reduce the Validation Dataset Size
- Increase batch_size

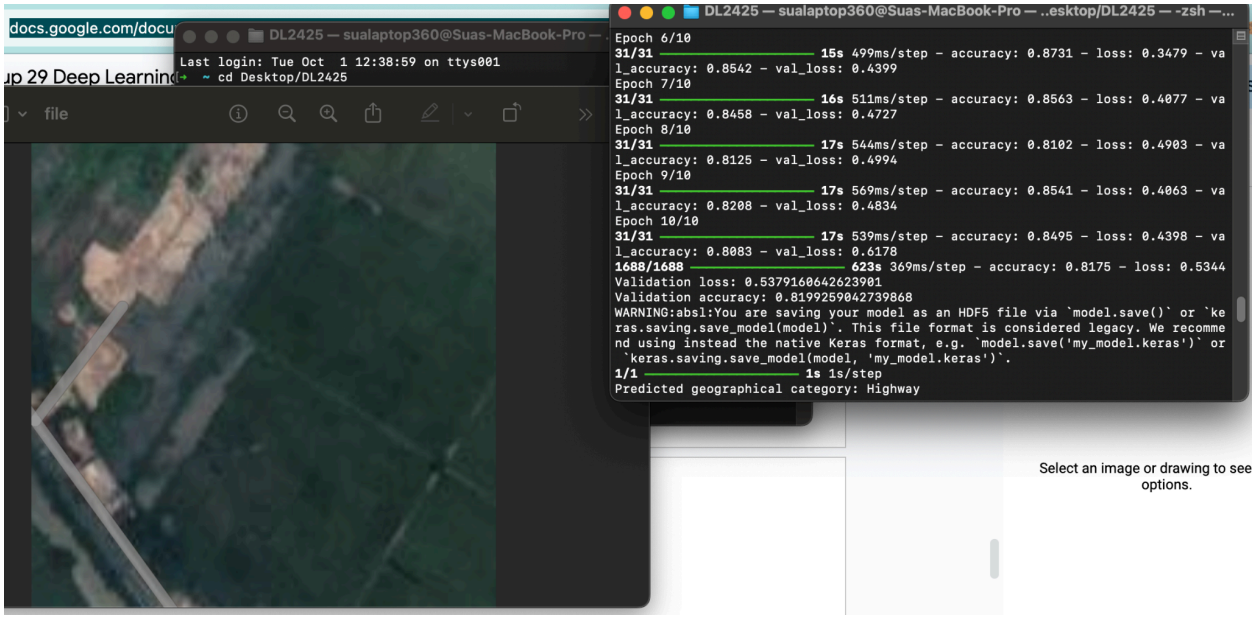


7. Experimental Setup

This section will detail the hardware and software environment used to train and test the MobileDet model. It will include:

- Hardware: Specifications of the machine or platform used (e.g., GPU, CPU, mobile device).
- Software: Frameworks and libraries used (e.g., TensorFlow, PyTorch, OpenCV).
- Training Configuration: Details on training duration, batch size, and any other relevant parameters like model checkpoints.

8. Experiment Results



8.1. Performance Metrics

After training, the model's performance is evaluated on the validation dataset. The following results were obtained:

- Validation Loss: 0.538
- Validation Accuracy: 0.82

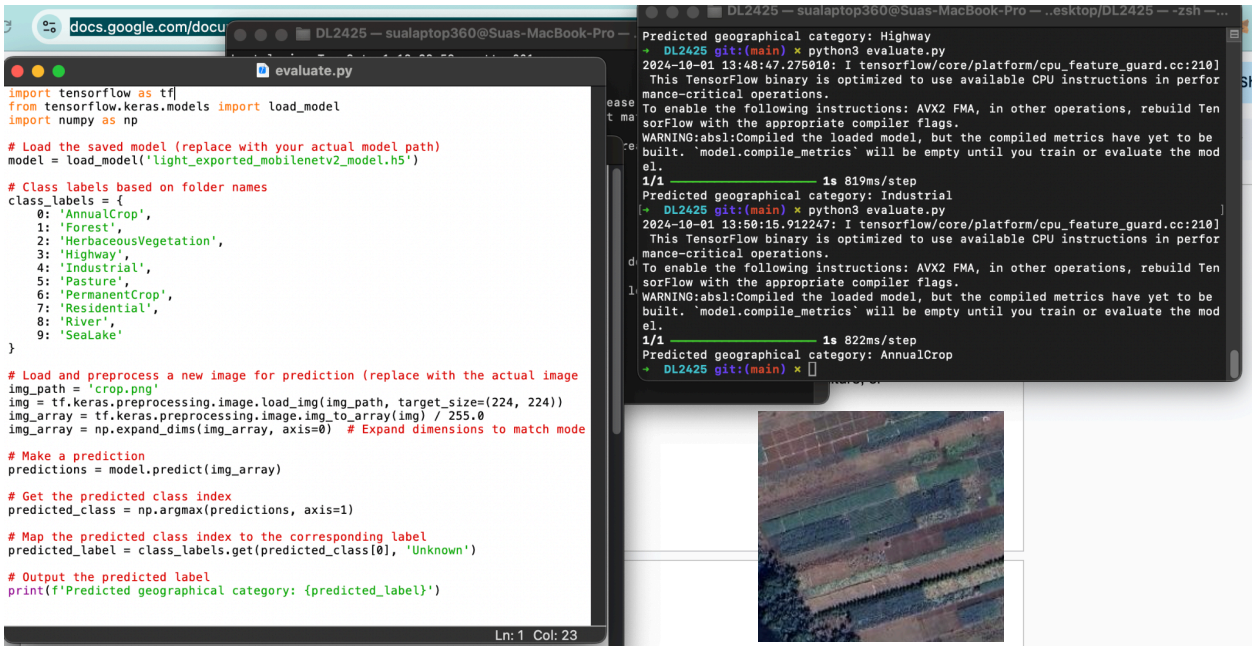
These metrics indicate that the model has learned to generalize well to unseen data.

8.2. Visualization of Training Progress

The training and validation accuracy and loss are plotted over the 10 epochs to visualize the learning curve. The accuracy increases steadily over the epochs, while the loss decreases, showing that the model is converging.

8.3. Prediction on New Data

The trained model is capable of making predictions on new remote sensing images. Given a new satellite image, the model classifies it into one of the categories: forest, agriculture, or urban…



9. Discussion

The discussion will analyze the results and offer insights into the strengths and weaknesses of MobileDet in the context of remote sensing. Key points will include:

- Performance Analysis: How well MobileDet handles large, high-resolution remote sensing images compared to other models.
- Challenges: Issues such as small object detection,

- Adaptability: Discussion on the model's ability to adapt to different classes and scales of objects specific to remote sensing, like buildings, vehicles, and natural features.
- Limitations: Possible drawbacks such as reduced accuracy under certain conditions or complexity in fine-tuning for specialized tasks.
- Comparison with State-of-the-Art: A critical comparison with other lightweight and full-scale models in terms of accuracy-efficiency trade-off.

10. Optimization and Future Work

This section will explore potential improvements and optimizations to enhance MobileDet's performance in remote sensing scenarios, including:

- Architectural Modifications: Exploring if modifications like adding atrous convolutions or using spatial pyramid pooling can improve feature extraction for remote sensing.
- Transfer Learning: Using pre-trained models on similar tasks or large-scale datasets to boost initial performance.
- Hardware Acceleration: Investigating the use of specialized hardware (e.g., Tensor Cores, Edge TPUs) for faster inference.
- Future Directions: Suggestions for future research, including multimodal integration (e.g., combining SAR and optical images), and leveraging advanced techniques like attention mechanisms.

11. Conclusion

The conclusion will summarize the study's main findings, emphasizing MobileDet's potential as an efficient solution for object detection in remote sensing. It will reiterate the significance of the work in enabling real-time, resource-constrained applications in fields like disaster management and urban planning. Additionally, it will highlight the importance of ongoing research in refining lightweight models for complex remote sensing tasks.

12. References

Keras 3: Deep Learning for Humans :https://github.com/keras-team/keras
EuroSAT dataset: https://github.com/phelber/EuroSAT
Model: *https://www.kaggle.com/models/google/mobilenet-edgetpu-v2/tensorFlow2*