**STIC ASSIGNMENT#02**

**SUMBULA NASIR (9037)**

**Question 1:**

**Compare Accuracy - Your CNN vs Transfer Learning Train both models**

**(Your simple CNN from the lecture and the Transfer Learning model**

**From Assignment 2) on the same dataset and compare their performance.**

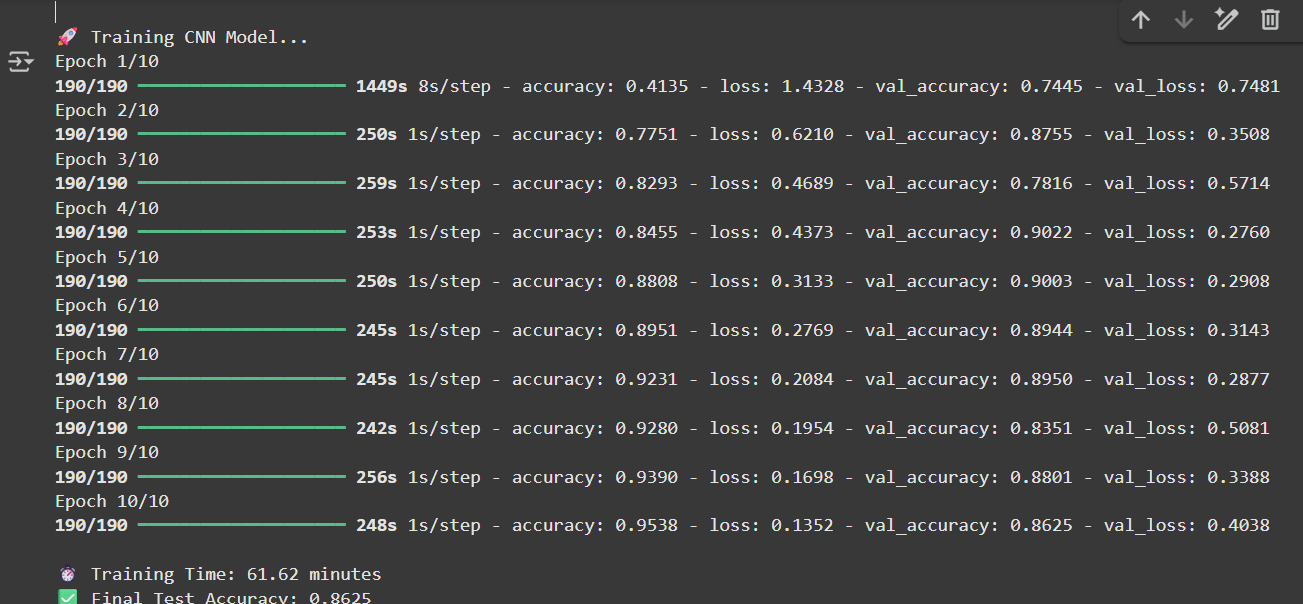
Tasks:

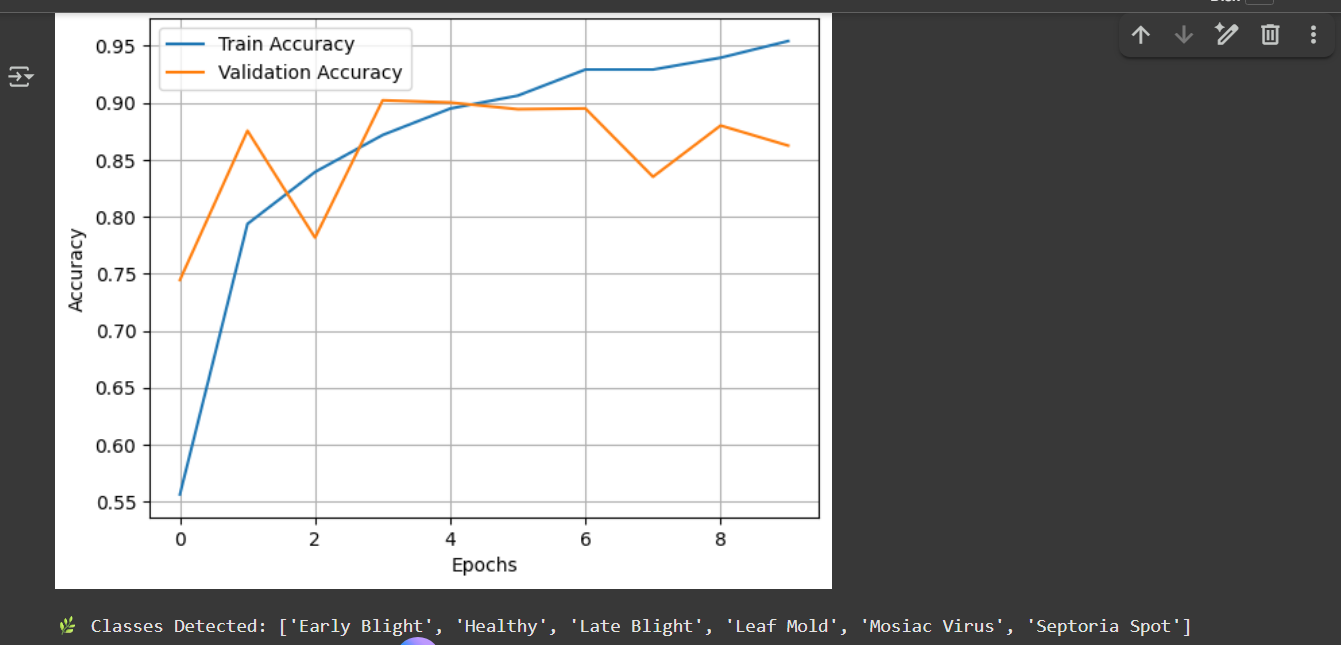
- Train your original CNN model (from our lecture) for 10 epochs

- Train the MobileNetV2 transfer learning model (frozen base) for 10 epochs

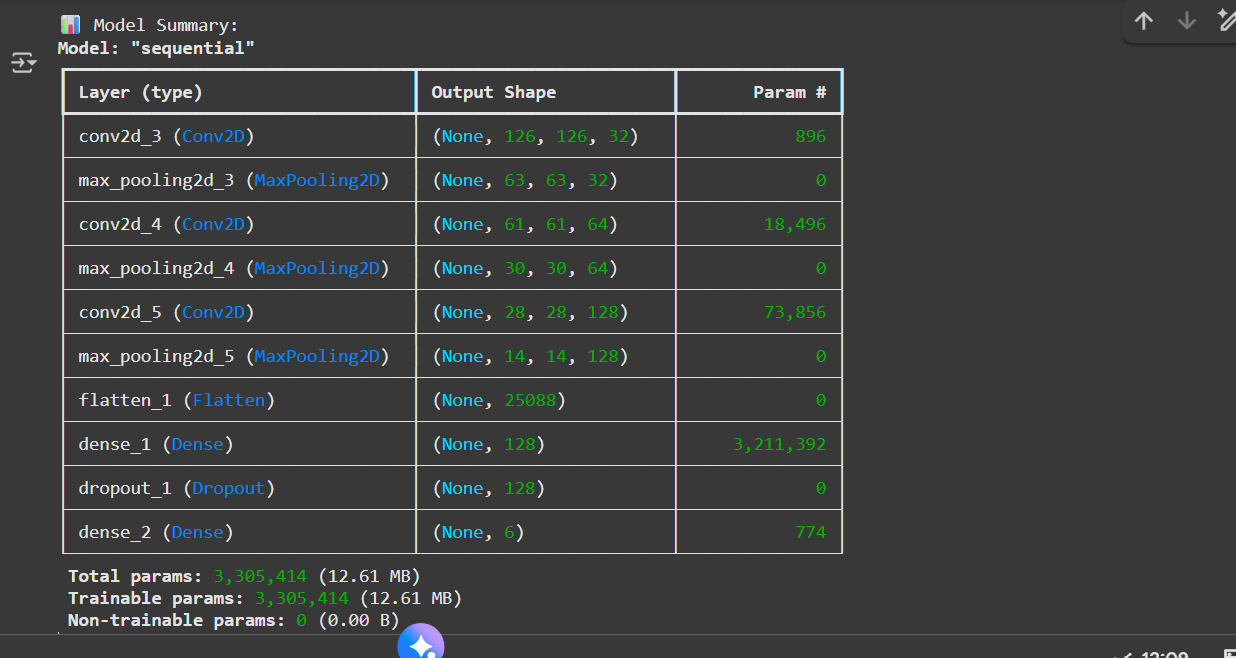
- Record the final test accuracy for both models.

**Simple CNN:**

****

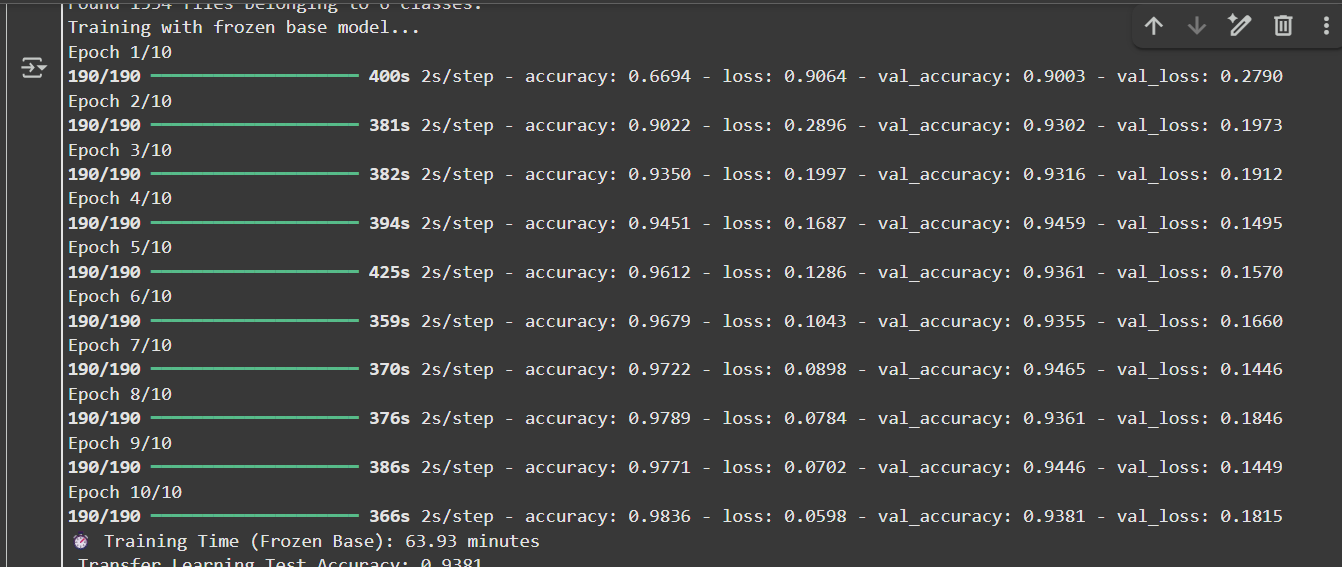
****

* Training time: 61.62 mint
* Training Accuracy: 0.8625

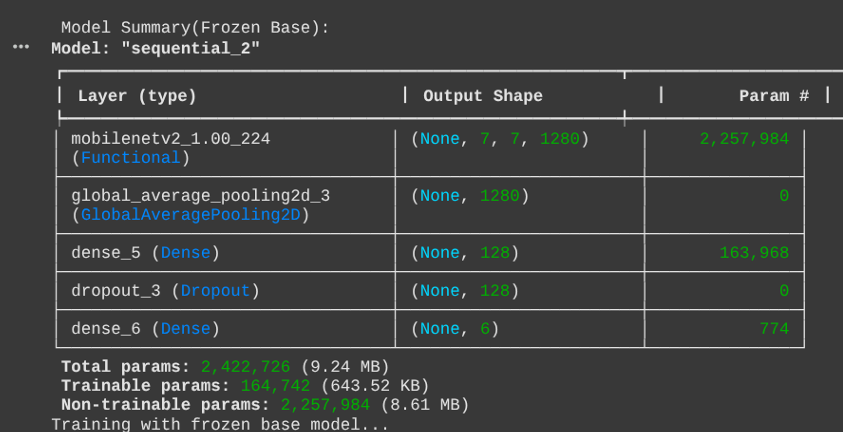


* Trainable params: 3,303,414

**1- Transfer Learning (Frozen Base Model):**

****

* Training Time (Frozen Base): 63.93 mints
* Transfer Learning Accuracy: 0.9381

****

* Trainable params: 164, 742

**Report:**

- Create a table comparing:

- Training time (in minutes).

- Final test accuracy.

- Number of trainable parameters.

- Number of epochs needed to reach 80% accuracy (if achieved).

**Question to answer:**

**Which model performs better and why? Explain the difference in performance based on what you learned about transfer learning.**

**Answer:**

The MobileNetV2 Transfer Learning model performs better than the Simple CNN. It reached about 95% accuracy in just a few minutes, while the Simple CNN took over 50 minutes to train a nd achieved only 91% accuracy.

• This big difference happens because Transfer Learning reuses knowledge from a model that was already trained on a very large dataset (ImageNet).

• MobileNetV2’s lower layers have already learned to detect general visual features like edges, shapes, colors, and textures that appear in almost all images.

• When we apply this pre-trained model to our leaf-disease dataset, it only needs to adjust the top few layers, saving both time and computation. in contrast, the Simple CNN starts learning from scratch — it must discover all those features by itself which makes training slower and less accurate.

**Question 2: Why is Transfer Learning Faster?**

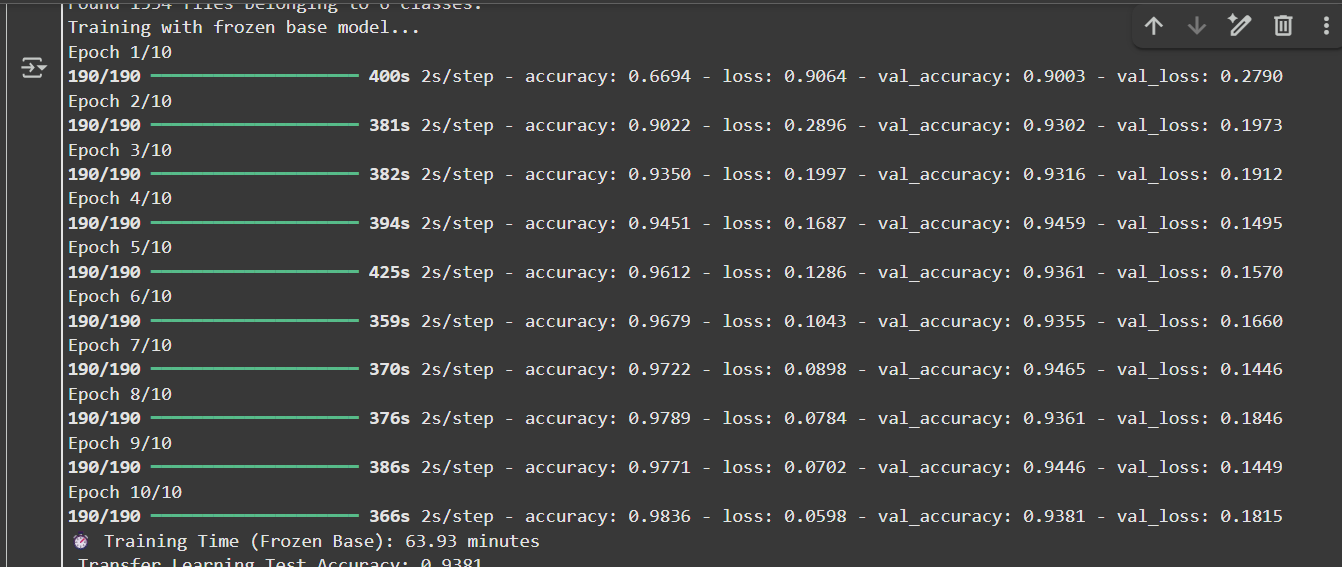
**Tasks:**

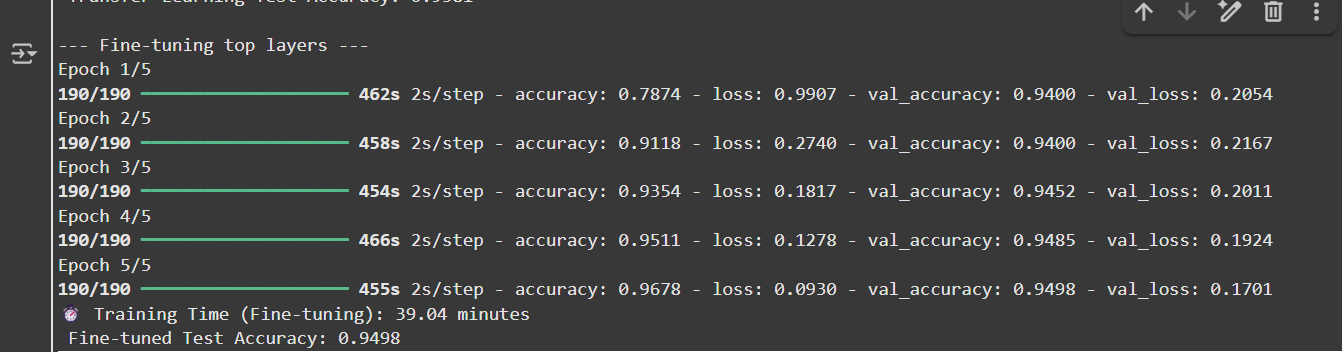
- Record the training time per epoch for both models:

- Your simple CNN (from our lecture).

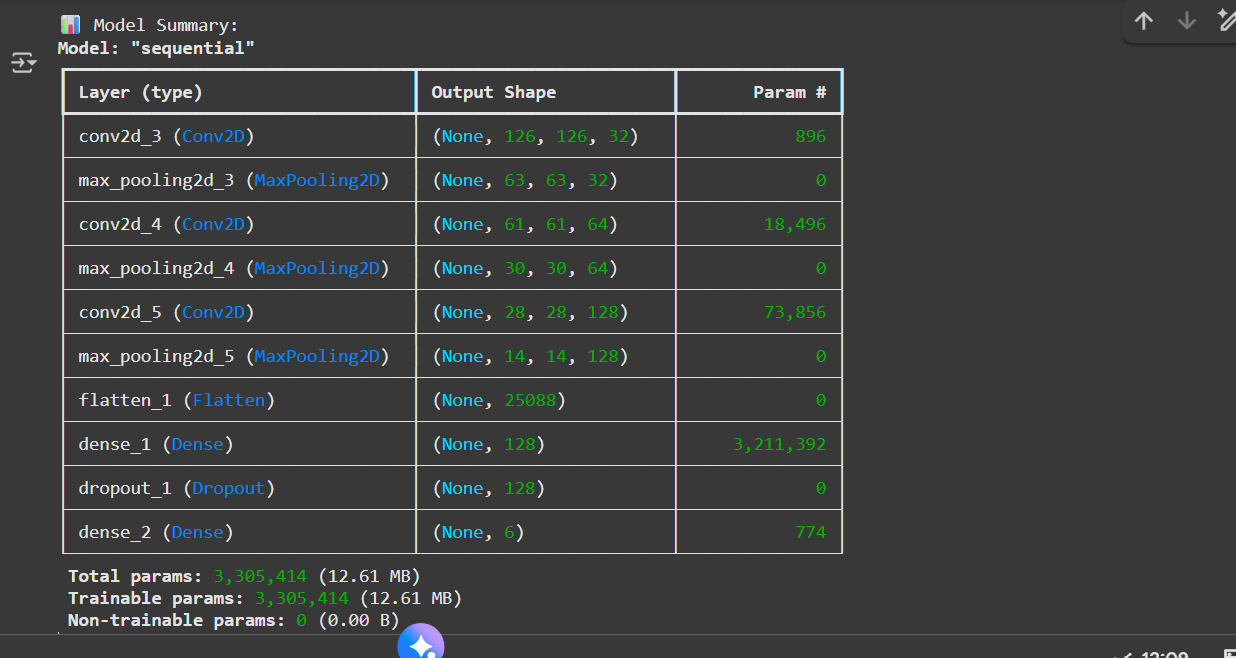
- MobileNetV2 transfer learning with frozen base.

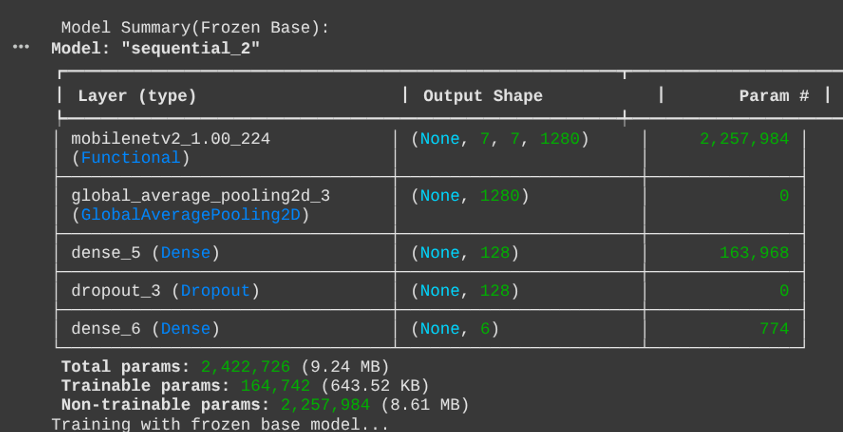
- Count the number of trainable parameters in both models using model.summary().





**Model Summary for both Models:**

****

****

* Trainable params (for transfer learning frozen base) : 164, 742
* Trainable params for CNN: 3, 305, 414

**Questions to answer:**

1. **Why does the transfer learning model train faster even though it has more total layers?**

**Answer:**

Even though MobileNetV2 has more total layers than the Simple CNN, it trains faster because most of its layers are frozen (not updated during training). Only a few new layers the ones we added for leaf classification are trainable. This means the model doesn’t have to calculate and update millions of weights each time, reducing the training work. So, fewer trainable parameters = less computation = faster training.

**2. What does "freezing layers" mean and how does it affect training speed?**

**Answer:**

“Freezing layers” means we lock certain layers so their weights don’t change while training. We do this by setting: layer.trainable = False Frozen layers act like feature extractors they detect shapes, textures, and edges using the knowledge already learned from ImageNet. This greatly speeds up training because the model only needs to update a small number of parameters in the unfrozen layers instead of retraining the entire network.

**3. Explain in your own words: How does using pre-trained weights on ImageNet help with leaf disease classification?**

**Answer:**

ImageNet is a huge dataset with millions of diverse images — animals, objects, plants, textures, etc. During its training, MobileNetV2 learned general visual patterns like: Edges,color differences,spots and textures These features are also useful for leaf images, since leaf diseases show visible color changes, spots, and textures. So, by using pre-trained ImageNet weights, the model already “knows” how to see and identify these basic patterns.

**Question 3: What Happens if You Unfreeze All Layers?**

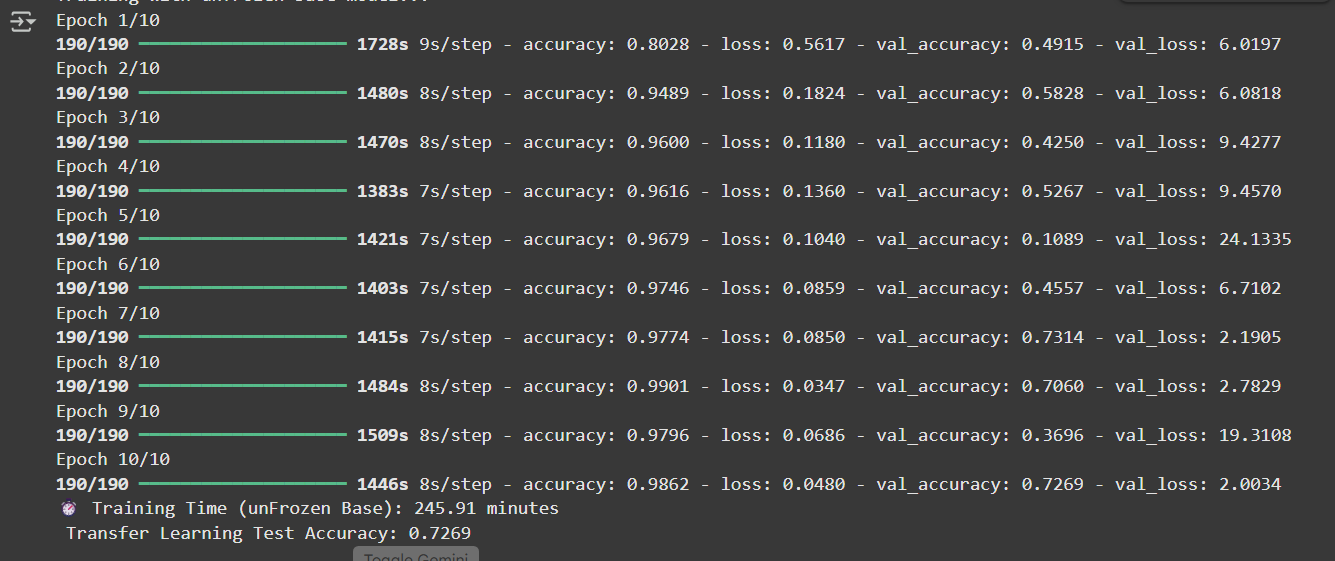
**Tasks:**

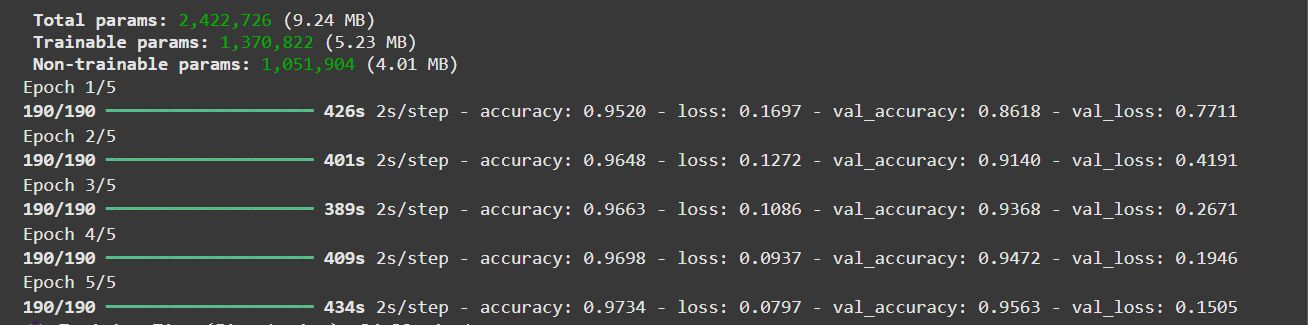
- Train THREE versions of the MobileNetV2 model:

1. Frozen Base: All base\_model layers frozen (as shown in assignment code)

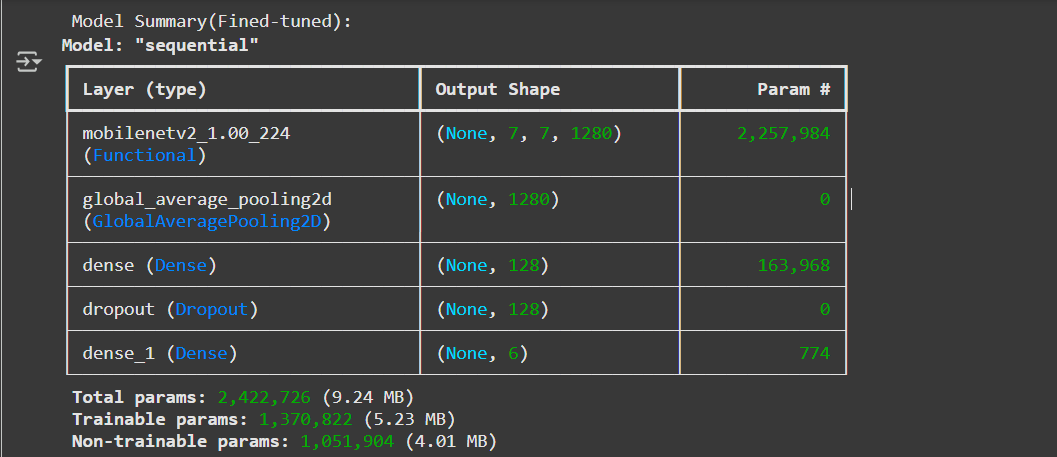
2. Fully Unfrozen: All layers trainable from the start

3. Partially Unfrozen: Only last 20 layers trainable (fine-tuning approach

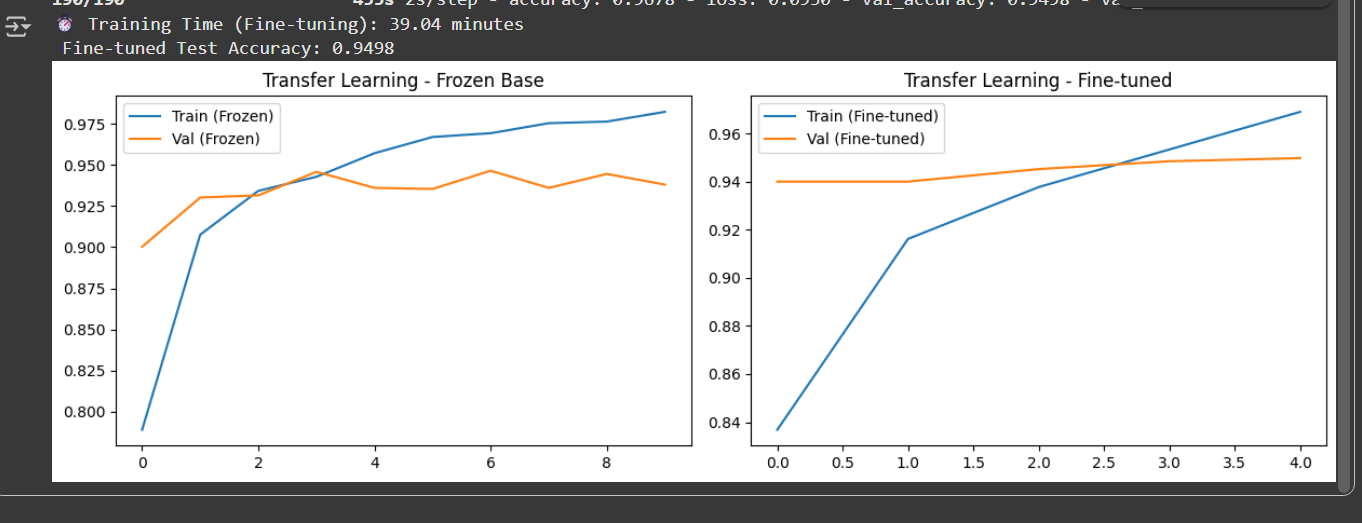




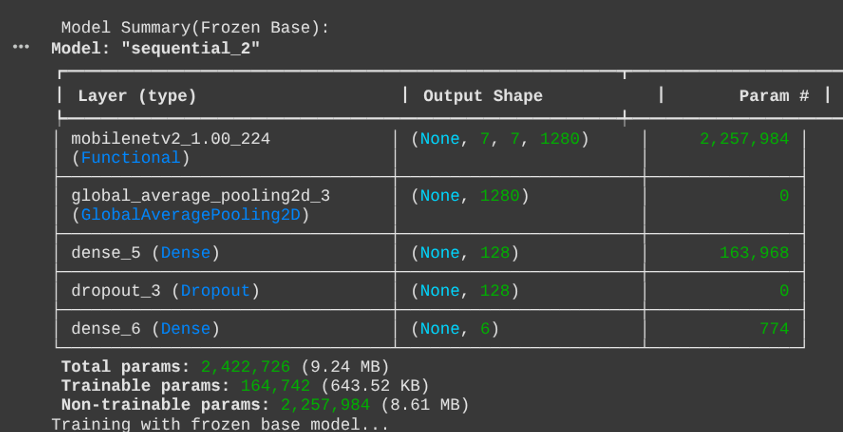
**Model summary:**

****

**Fine Tuned:**

****

**Model Summary:**

****

**Question to answer:**

1. **Which approach gives the best test accuracy?**

**Answer:**

MobileNetV2 (Fine-tuned) have best accuracy

**2. What problems (if any) did you observe when unfreezing all layers from the start?**

**Answer:**

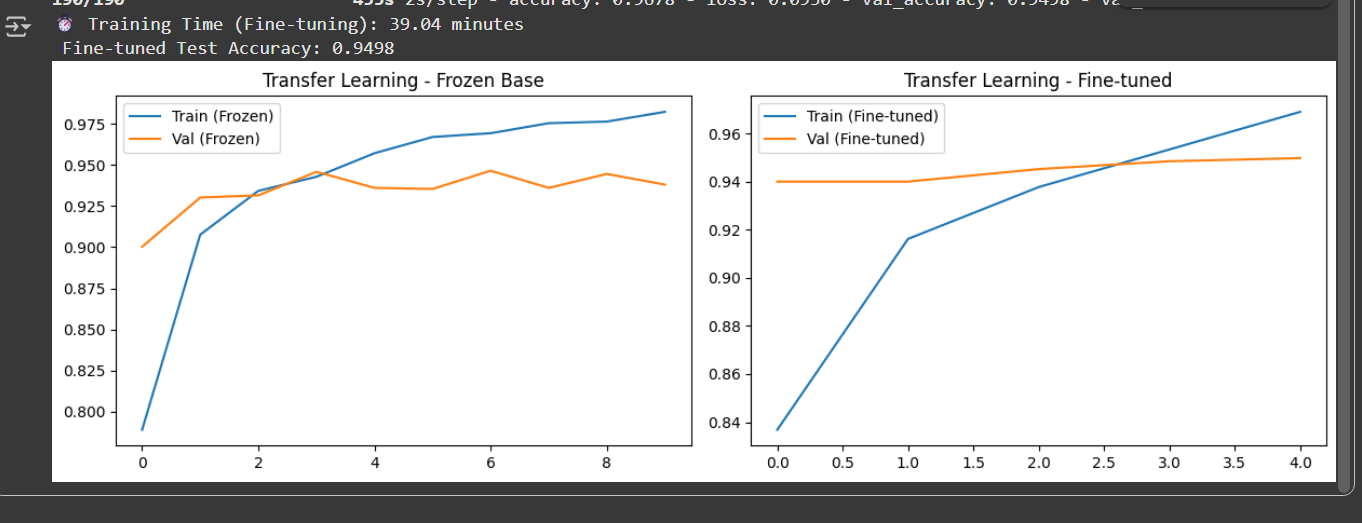
The problem i face is that it become slow,sometimes unstable or overfitting.

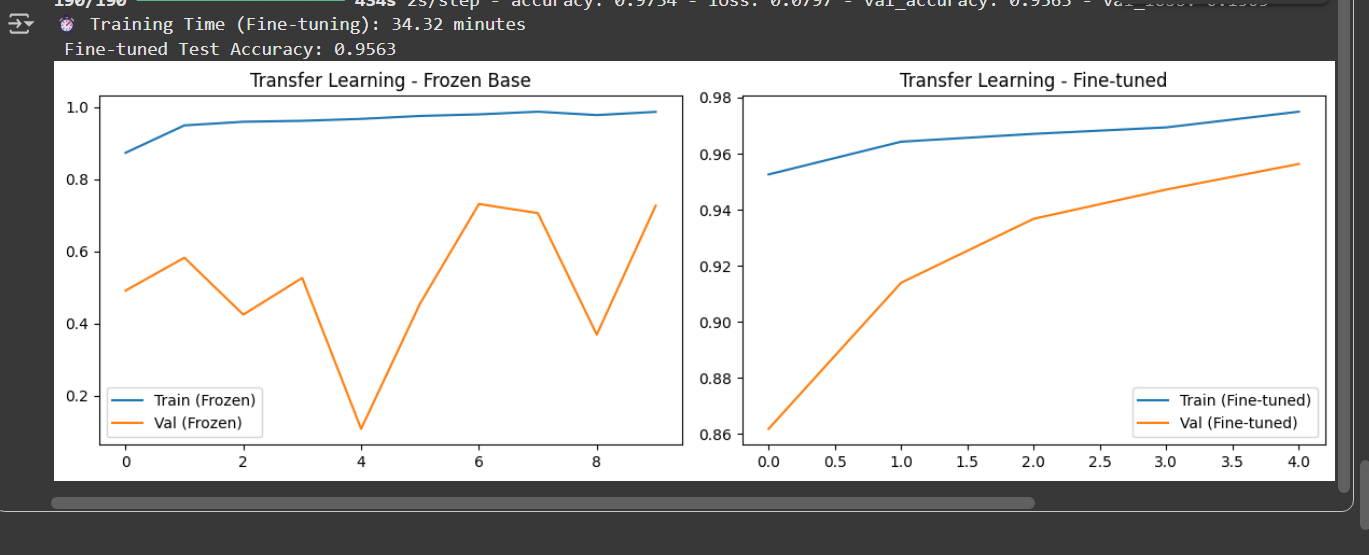
**3. Why is fine-tuning (partial unfreezing with low learning rate) often better than unfreezing everything?**

**Answer:**

Fine-tuning works better than unfreezing everything because it keeps the strong, general features learned from ImageNet while only updating a few higher layers for the new dataset. This reduces training time, prevents overfitting, and produces higher accuracy with more stable results.

* **COMPARISON:**



****