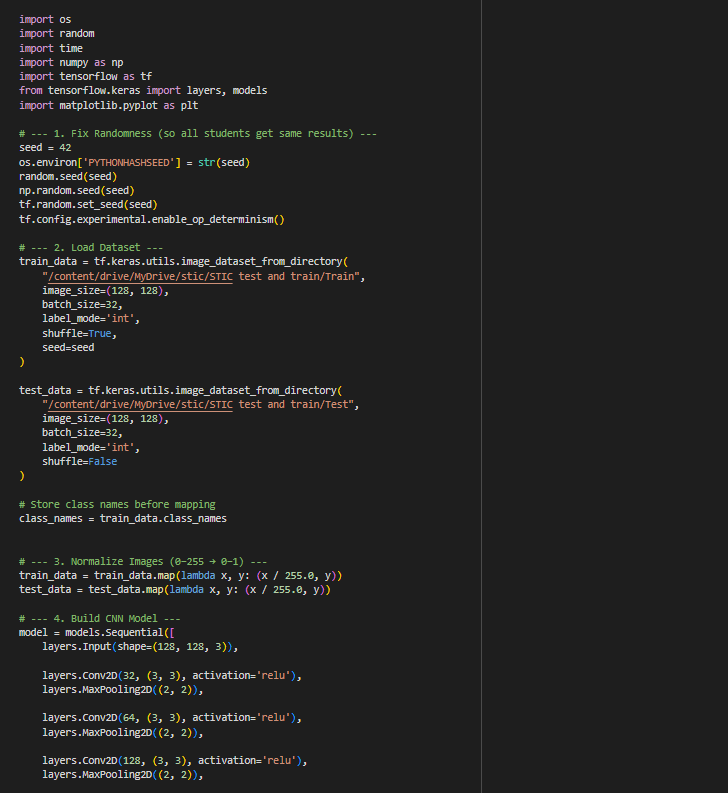
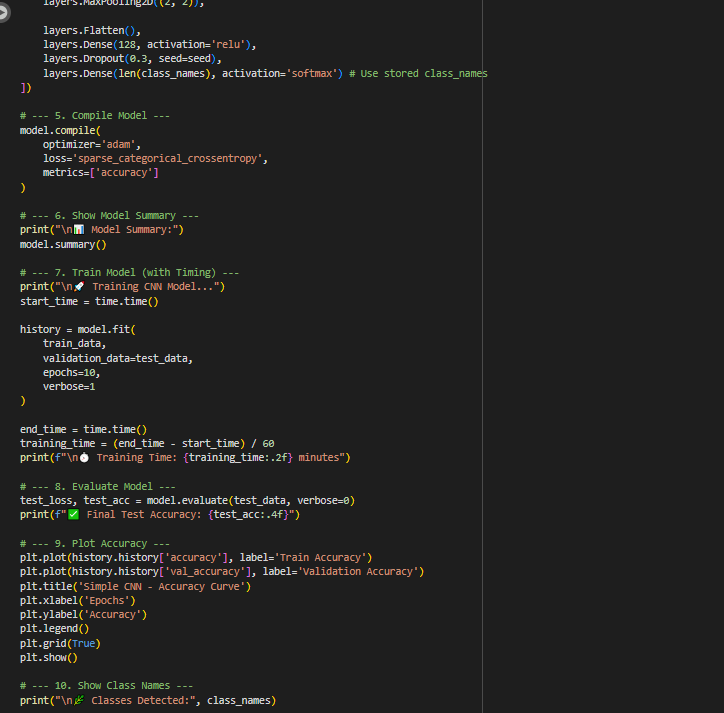
**NAME :** Muqadas Israr

**Reg # :**  9011

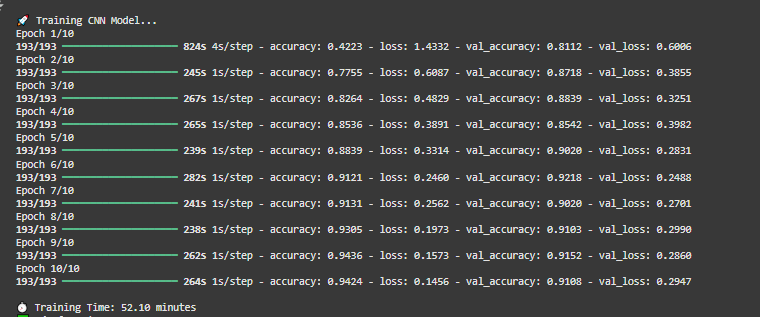
**Cnn**

Code # 01

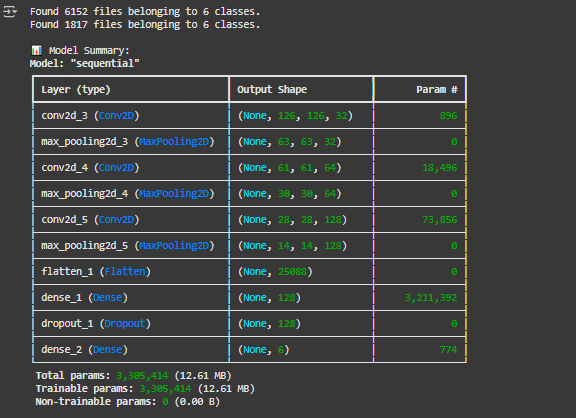


-

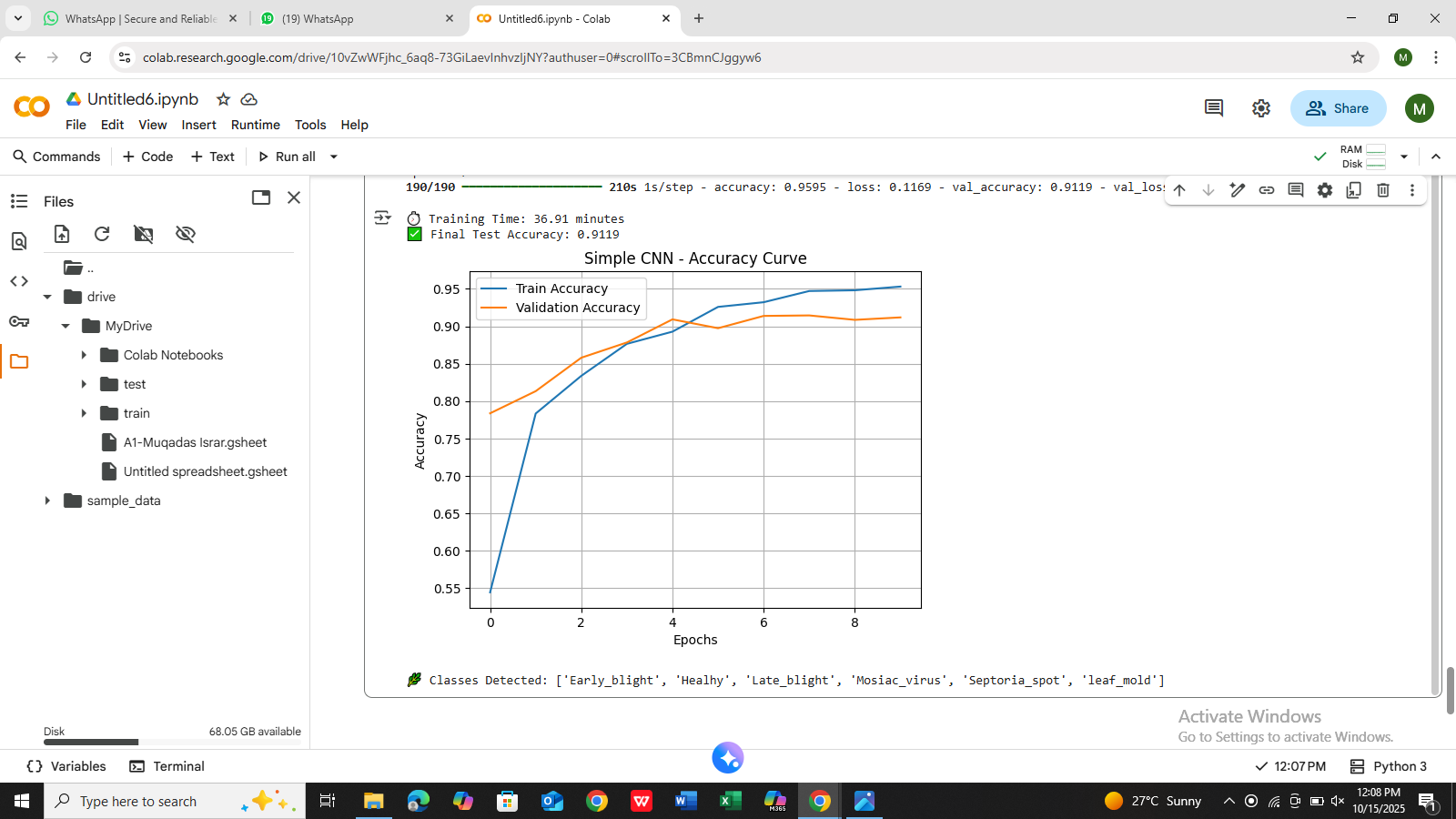
**Epochs**



**Summary:**



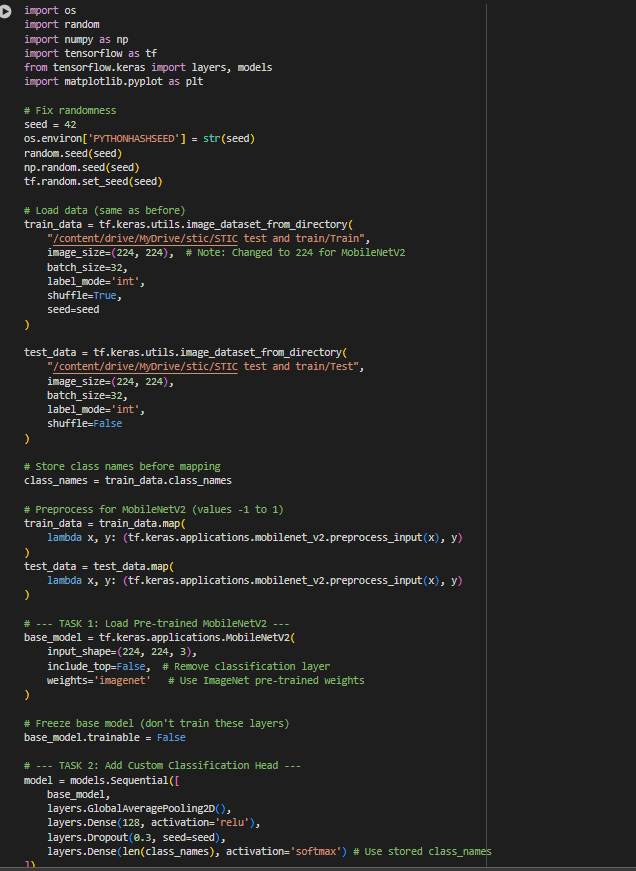
**Graph**

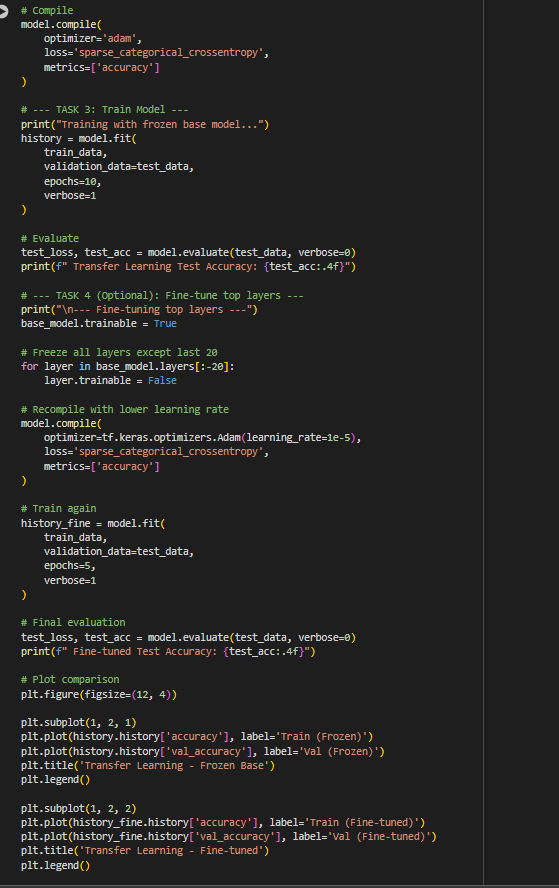


**Code #2**

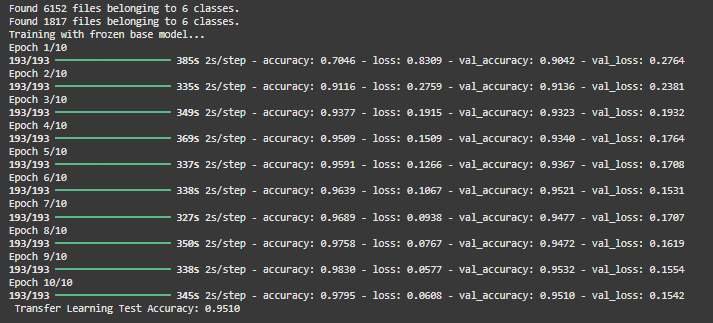
Transfer Learning

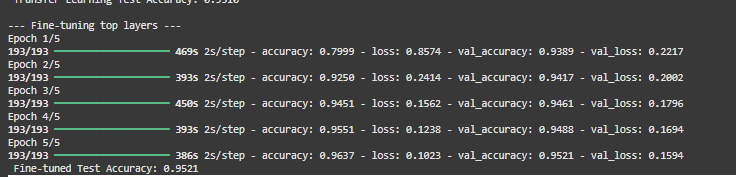
**CODE:**



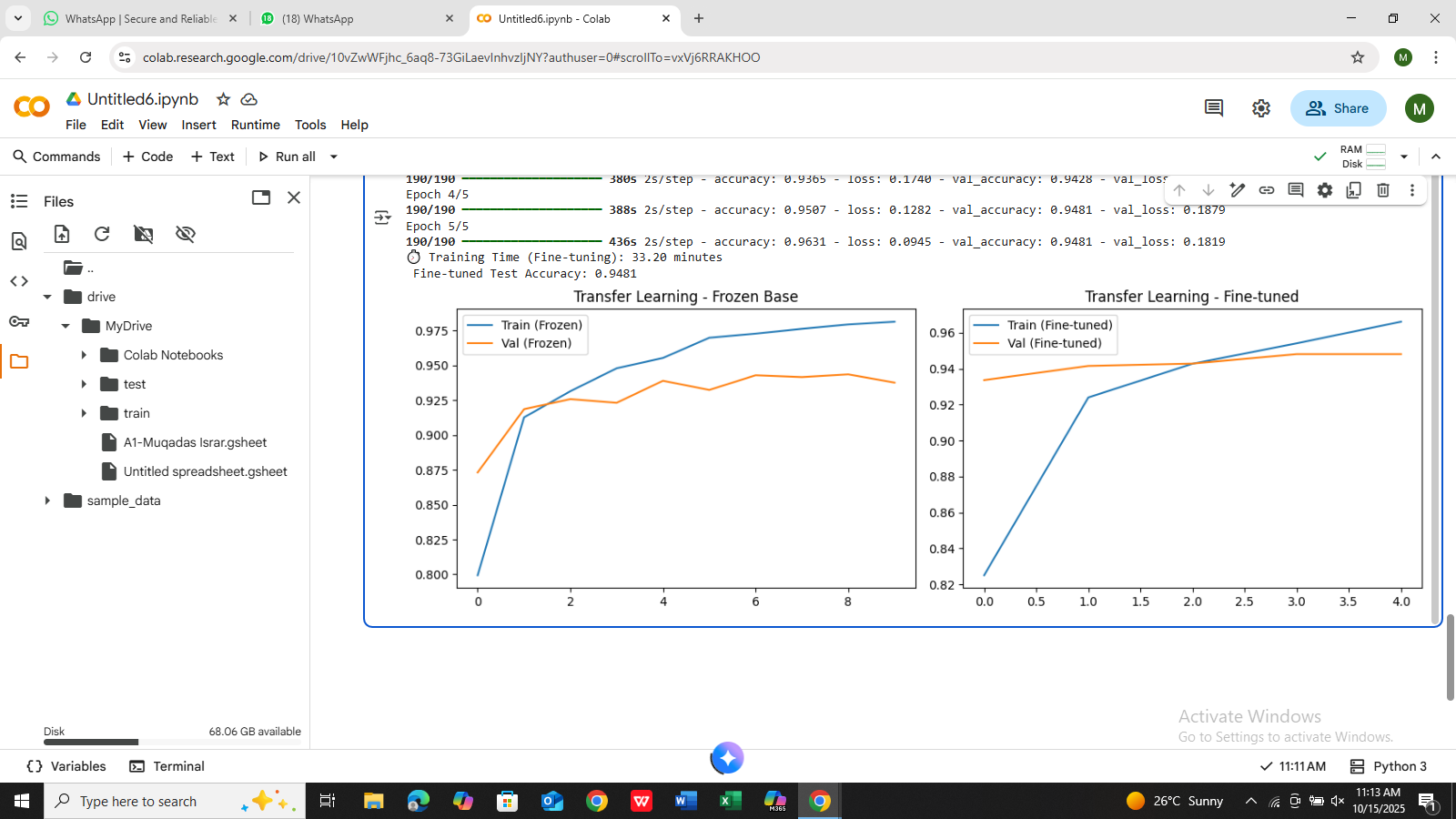


**Epochs:**





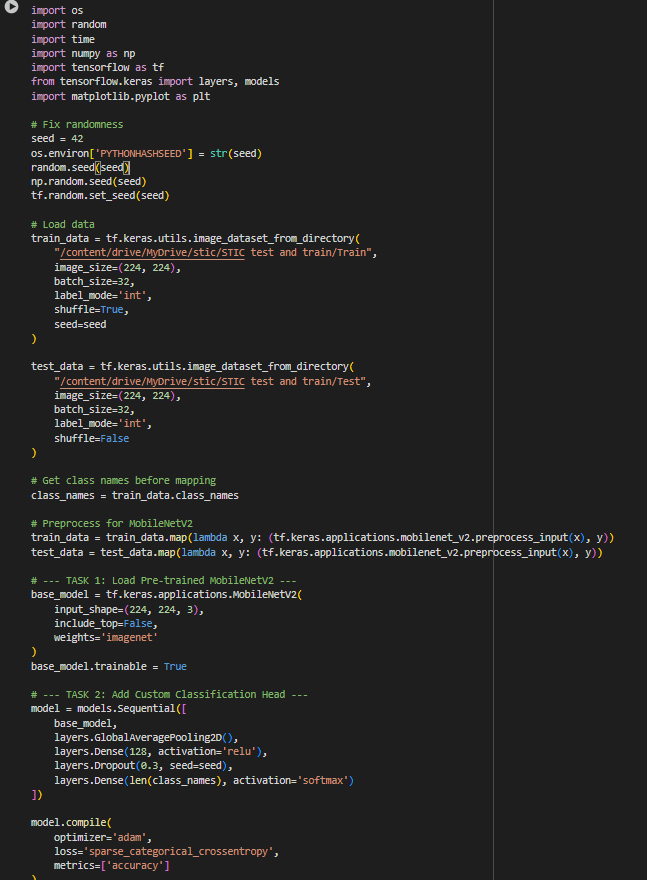
**Graph**

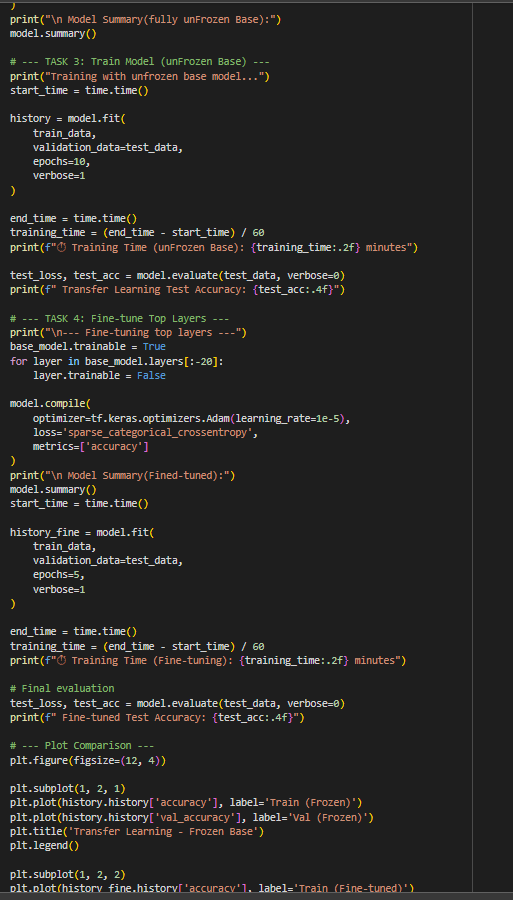


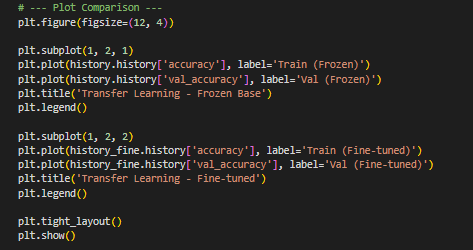
**Code# 03**

MobileNetV2 variants

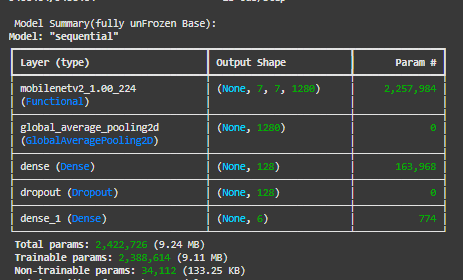
CODE







Table



Epochs

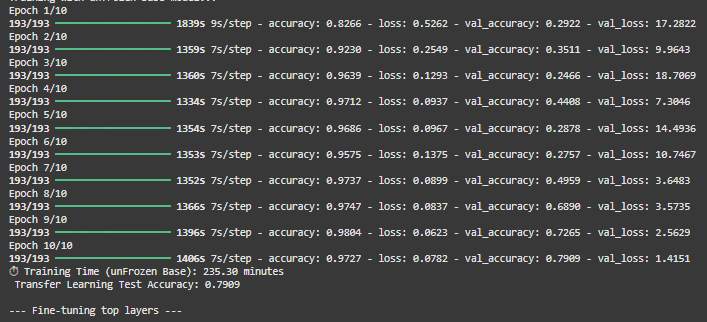
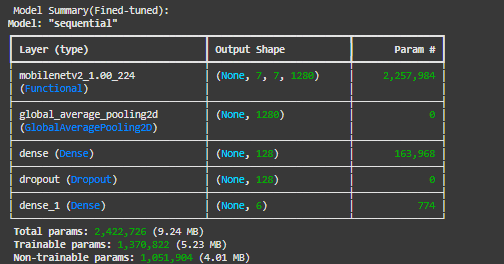
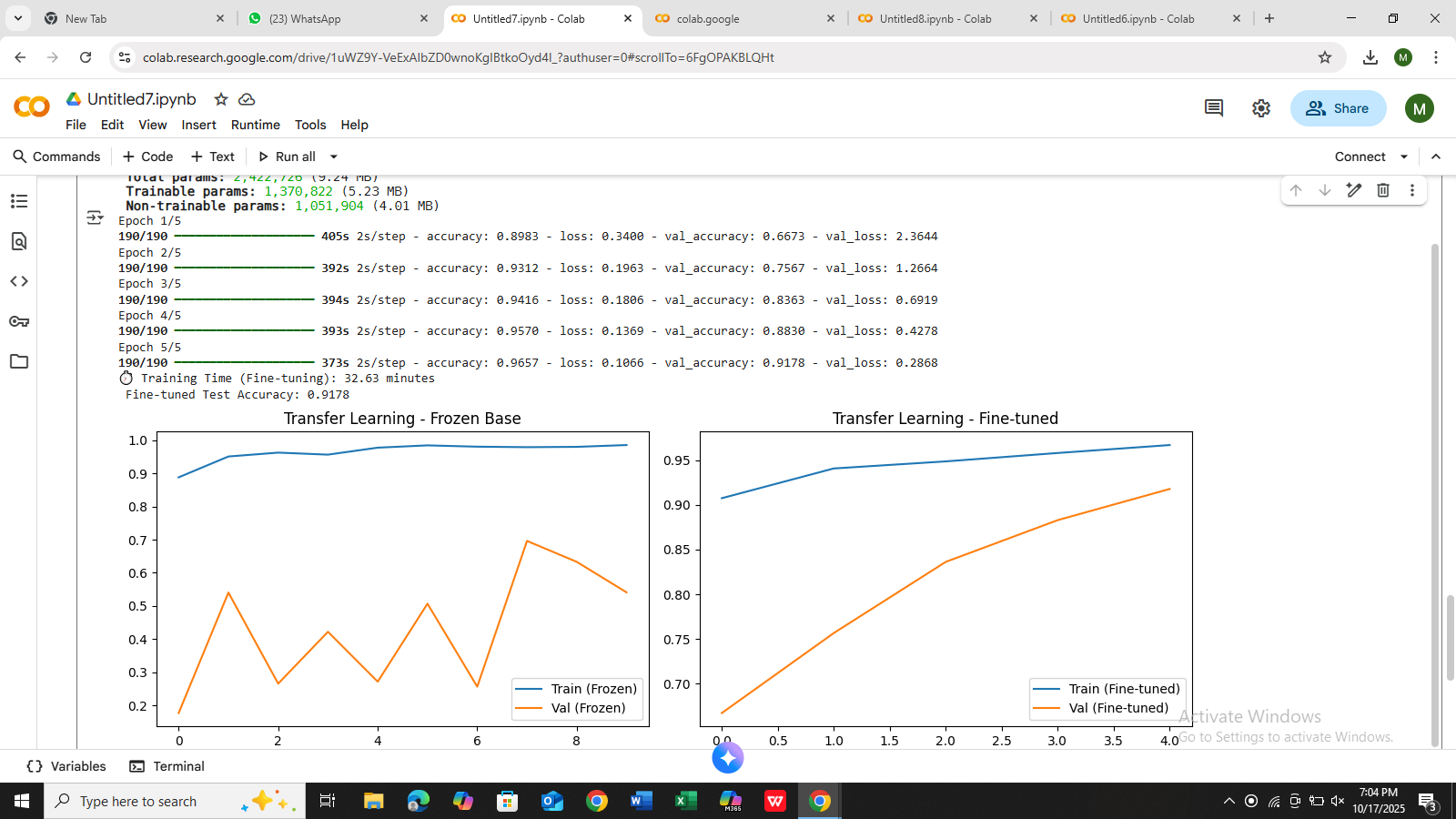


Table after epoch



**GRAPH**

****

**Question1:Compare Accuracy-Your CNN vs Transfer**

**Learning :**

**Model**

**Final Test**

**Accuracy**

**Training**

**Time**

**Trainable**

**Parameters**

**Epochs to Reach**

**80% Accuracy**

**Simple CNN**

**0.923**

**45.74 min**

**3,304,414**

**3 epochs**

**MOBILE NET V**

**(Frozen Base)**

**0.93**

**64.99 min**

**164,724**

**2 epochs**

**MOBILE NET V**

**(Fine-tuned)**

**0.79**

**33.15 min**

**1,370,822**

**1 epochs**

**Question 2: Why is Transfer Learning Faster?- Why does the transfer learning model train faster**

**even though it has more total layers?**

It is so as most of the layers in pretrained model has frozen (not updated)

weights. Frozen layers do not calculate backprop updates (they still do the

forward pass but they don't run backprop weight updates), so very few

parameters need to have gradients computed and weights updated. Less

trainable params → less gradient computations → faster training per epoch

and quicker convergence.

**- What does "freezing layers" mean and how does it**

**affect training speed?**

A layer freeze is a set layer. trainable = False. The weights of these layer

are not updated during training. This is to save both memory and

computation during backpropagation and not interfering with the features

learned in early layers. Therefore, it is much faster and more stable to train

with these good generic features preserved.

**- Explain in your own words: How does using pr**

**etrained weights on Image Net help with leaf disease**

**classification?**

The p retrained weights already captured these useful low-level features

(edges, corners, color blobs) and mid-level texture/shape detectors that

generalize to many vision tasks. The low-level visual features in images of

leaf disease are indeed imbued by the same kinds of characteristics— so

we don’t need our network to learn them from scratch; it only needs to be

equipped with learning combinatorial task-specific higher-level

representations. This accelerates convergence and typically improves final

accuracy -- especially for small datasets.

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

The Frozen Base model trained fast but showed unstable validation

accuracy around 0.6–0.7 → underfitting

The Partially Unfrozen (fine-tuned) model gave better and smoother

validation accuracy (0.7935) → more stable and accurateIf all layers were unfrozen, training would become slow and unstable, and

the model would likely overfit and lose the pretrained features, causing

lower accuracy.

So, according to your graphs, partial fine-tuning gives the best and most

stable performance, while fully unfreezing all layers would hurt accuracy.

**Questions to answer:**

**- Which approach gives the best test accuracy?**

The Simple CNN model gives the best test accuracy — 0.9235.

It achieved a final test accuracy of 0.9235, with smooth and consistent

training and validation accuracy curves.

**What problems (if any) did you observe when**

**unfreezing all layers from the start?**

Training became slow and unstable.

The model started over fitting training accuracy high, validation low).

It lost pretrained features, so performance got worse instead of better.

**- Why is fine-tuning (partial unfreezing with low**

**learning rate) often better than unfreezing everything?**

Because it keeps useful p retrained features from Image Net while allowing

the top layers to learn task-specific details.

It trains faster and more stable, with less over fitting.

A low learning rate updates weights gently, improving accuracy without

destroying p retrained knowledge.