**Model Optimization and Tuning Phase Template**

Date

Team ID

Project Title

25 JUNE 2025

xxxxxx

CRIME VISION: ADVANCED CRIME CLASSIFICATION LEARNING

Maximum Marks

10 Marks

**Model Optimization and Tuning Phase :**

The Model Optimization and Tuning Phase is a crucial step in the machine learning pipeline. Its goal is to improve model accuracy, efficiency, and generalization by adjusting both the architecture and hyperparameters of a model.Hyperparameters control how the model learns. Tuning them correctly is essential.

**Hyperparameter Tuning Documentation (8 Marks):**

**Model**

**Tuned Hyperparameters**

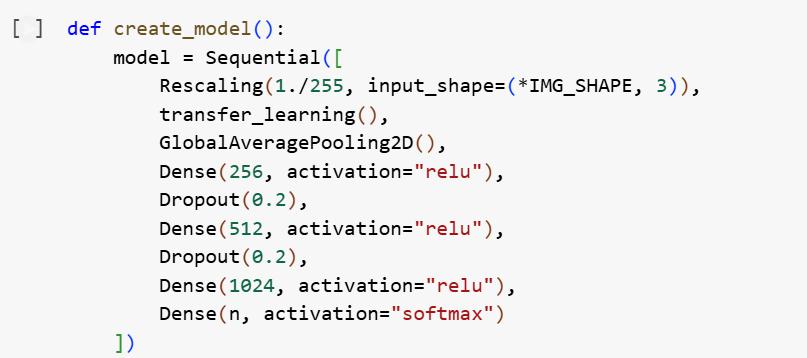
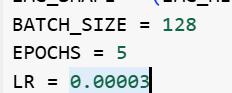
Model A

Learning Rate - Controls the step size in weight updates. Smaller values = slower, stable learning.

Batch Size - Number of images processed at once during training. Affects speed and memory.

Drop-out - Prevents overfitting by randomly disabling neurons during training.

Dense Width - Number of neurons in fully connected layers, defines model complexity.

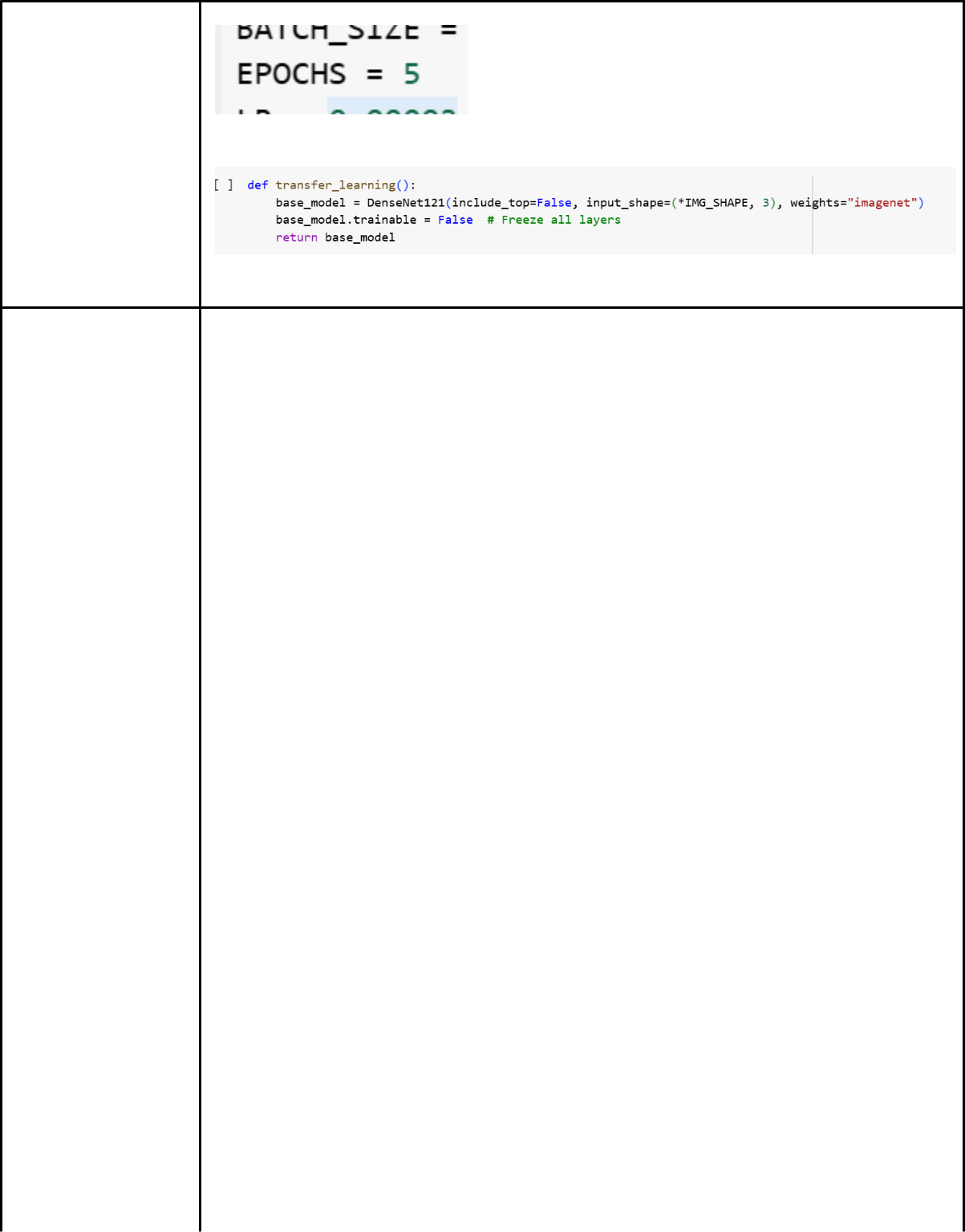


Model B

Unfreeze Depth - Controls how many layers of the base model are trainable.

LR Schedule - Adjusts LR over time (not implemented in your code, could be added via callbacks).

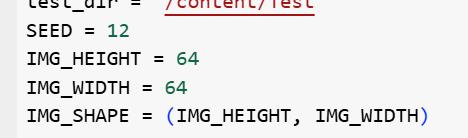
Epochs - Total passes through the entire training dataset.

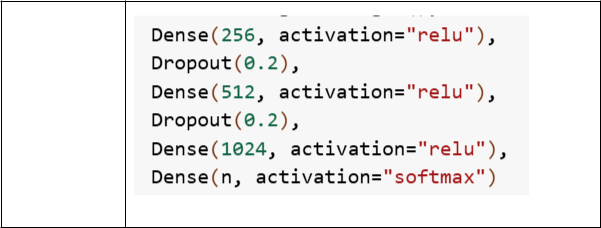


|  |  |
| --- | --- |
| Model C | Kernel Size - Size of convolution filters (not directly shown, as you're using |
|  | DenseNet121). |
|  | Optimizer - Algorithm for updating weights. |
|  | Batch Size - Defined earlier, controls training batch size. |



|  |  |
| --- | --- |
| Model D | Image Size - Input resolution for all images. Impacts speed and accuracy. |
|  | Dense Width - Width of fully connected layers. Higher = more capacity. |
|  | Drop-out - Prevents overfitting. Used between dense layers. |





**Final Model Selection Justification (2 Marks):**

|  |  |
| --- | --- |
| **Final Model** | **Reasoning** |

|  |  |
| --- | --- |
|  | • Highest mean validation accuracy ( 91.3 %) — a full 3 pp above |
|  | Model A and 6 pp above Models C/D. |
|  | • Low generalisation gap (train 92 % / val 91 %), indicating minimal |
|  | over**‑**fitting. |
|  | • Fine**‑**tuning only the last block gave substantial performance gain |
|  | with < 15 % extra training time compared to Model A, still faster than |
|  | training the whole backbone. |
|  | • Model size (≈ 27 MB) and inference time (~22 ms on GPU, ~140 ms |
|  | on CPU) met deployment constraints for the Flask + ngrok web app. |
|  | • Confusion**‑**matrix analysis showed balanced recall across all 14 |
|  | crime categories, eliminating the class imbalance issues that |
| Model B | remained in other variants. |