Artificial Neural Networks Project

Konstantina Ellina

June 20, 2024

Training settings

At first, the fully-supervised learning scheme was used to train a convolutional neural network for scene classification. The model chosen for this task is EfficientNet-B0, pre-trained on the ImageNet dataset. Secondly, the self-supervised learning scheme was used to design 2 pretext tasks(one for blurring and one for perturbation) used for scene classification. The models were fine-tuned on the 15-Scene dataset. The data is divided into training and test sets using stratified sampling to ensure an equal distribution of classes. The settings below are the ones used for the training of the models.

Table 1: Training Settings for Fully-Supervised Learning

Learning Rate	**Batch Size**	**Optimizer**	**Number of Epochs**	**Fully-Connected Layers**
0.001	32	Adam	20	1
0.001	64	Adam	20	1
0.01	32	Adam	20	1
0.01	64	Adam	20	1
0.1	32	Adam	20	1
0.1	64	Adam	20	1

Table 2: Training Settings for Gaussian Blurring + Black and White Perturbation

Learning Rate	**Batch Size**	**Optimizer**	**Number of Epochs**	**Fully-Connected Layers**
0.001	32	Adam	16	1

Learning Rate
Learning rates of 0.001, 0.01, and 0.1 were tried to explore a range of gradient descent steps. Lower learning rates help in gradual convergence, reducing the risk of overshooting minima, while higher rates speed up training but may risk instability. With trying 3 of them, I would see how this trade-off works and which is the best option for my data.

Batch Size
Batch sizes of 32 and 64 were chosen to balance memory constraints and training efficiency. Smaller batch sizes provide a more stochastic gradient descent, while larger batches offer more stable gradient estimates. Again, by trying various batch sizes we would be able to see this trade-off that is implying here and decide in the best parameter in the end. Also, batch size of 128 was tried, but this value is too high causing the GPU to crush, so I didn't use it in my last runs.

Optimizer
The Adam optimizer is selected for its adaptive learning rate capabilities, which help in efficiently navigating the loss landscape. This makes Adam particularly effective in handling sparse gradients and varying learning rates, leading to efficient and reliable convergence.

Number of Epochs
A fixed number of 20 epochs was used for all experiments to ensure sufficient training iterations while keeping computational costs reasonable.

Fully-Connected Layers
The EfficientNet-B0 model has a single fully-connected layer at the end, which was kept to match the pre-trained architecture and ensure compatibility with the fine-tuning process.

After all the experiments with the different parameters in fully-supervised learning, the best model was found. We count the best model as the one with the highest validation accuracy. I kept the parameters of the best model and trained from the start a model with these parameters and until the epoch that was found to have the highest accuracy. The parameters that got me the best model were used for the self-supervised models and was trained for 16 epochs that was the number of epochs with the best accuracy for the fully-supervised model.

Performance of models

To present the performance of the models in terms of classification accuracy, we summarize the results from the training logs for each model individually. As it was mentioned before, the models were trained in 16 epochs, which was the number of epochs that the fully-supervised model gave the best accuracy.

Fully-supervised learning model

In Table 3 below, we can see the training and validation performance over 20 epochs for the fully-supervised learning model, which gives us much information about the behavior of the model and its effectiveness in learning the scene classification task.

Training and Validation

The model shows rapid improvement in both training and validation accuracy during the initial epochs. The training loss decreases significantly and the validation accuracy improves steadily. Validation loss experiences fluctuations, indicating the model's attempt to generalize to unseen data and validation accuracy reaches 92.53% during this phase, demonstrating substantial learning from the training data. After the 10th epoch, the training loss stabilizes around a low value, indicating that the model has effectively learned the training data. Despite fluctuations, validation loss shows an overall trend towards stabilization, suggesting that the model is successfully generalizing. The highest validation accuracy of 92.75% is achieved at epoch 16 as shown in bold, confirming the model's strong performance. (The results can also be found in file "fully_supervised_0.001_and_32.txt".)

Table 3: Training and Validation Performance

Epoch	Train Loss	Validation Loss	Validation Accuracy (%)
1	0.7338	0.5383	84.84
2	0.3459	0.4527	88.18
3	0.2462	0.5111	87.40
4	0.2310	0.4791	89.74
5	0.1720	0.2547	92.53
6	0.1426	0.4571	90.08
7	0.1268	0.4407	90.41
8	0.1299	0.7283	91.08
9	0.1273	0.5439	87.96
10	0.1240	0.3453	90.64
11	0.1017	0.3760	90.52
12	0.1282	0.3585	91.30
13	0.1017	0.3373	90.86
14	0.0811	0.4551	87.96
15	0.1231	0.3862	90.86
16	**0.0835**	**0.3111**	**92.75**
17	0.0926	0.3890	89.19
18	0.0773	0.3280	92.08
19	0.0850	0.4360	90.41
20	0.1179	0.4054	92.20

Training and validation losses

The training and validation losses for the best model in fully-supervised model over several epochs are also displayed in the plot below. Although there is some instability in the model's validation performance, which is typical when training deep learning models, the overall trend indicates that the model's performance is improving on both training and validation sets. The training loss decreases steadily, indicating that the approach is mostly beneficial.

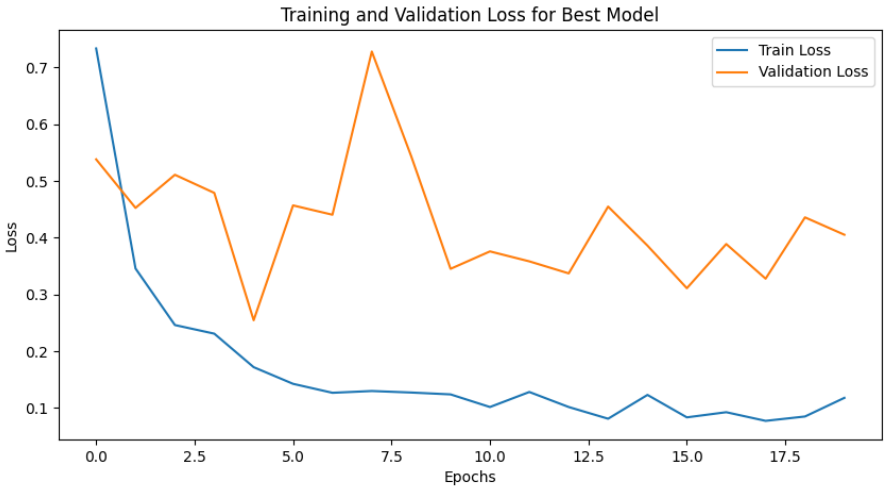

Figure 1: Training loss and validation loss for the best model

Test evaluation

The final evaluation on the test set indicates a robust performance of the fully-supervised learning model. The model achieves an overall test accuracy of 92.75%, demonstrating its capability to generalize well to unseen data. The precision, recall, and F1-score metrics are consistently high across all classes, indicating balanced performance. Highest precision (1.00) is observed in class 1 and class 14, indicating perfect positive predictive value. Highest recall (1.00) is seen in class 1 and class 6, suggesting the model correctly identifies all true positives in these classes. High F1-scores across all classes (above 0.85) show a good balance between precision and recall. Class 1 and class 6 have perfect precision and recall, showcasing the model's strong ability to distinguish these scenes. Class 0 and class 10 exhibit slightly lower precision and recall, indicating areas where the model could be further fine-tuned.

Table 4: Final Test Performance

Class	Precision	Recall	F1-score	Support
0	0.81	0.91	0.86	43
1	1.00	1.00	1.00	48
2	0.90	0.92	0.91	62
3	0.91	0.98	0.94	42
4	0.95	0.90	0.92	58
5	0.92	0.94	0.93	72
6	0.90	1.00	0.95	66
7	0.92	0.90	0.91	52
8	0.95	0.95	0.95	62
9	0.93	0.88	0.90	75
10	0.90	0.87	0.88	82
11	0.93	0.98	0.96	58
12	0.97	0.92	0.94	71
13	0.91	0.91	0.91	43
14	1.00	0.90	0.95	63
Accuracy		92.75%		

The confusion matrix highlights the model's effectiveness in correctly classifying most images, with minimal misclassifications. The diagonal elements represent the number of correct predictions for each class. The off-diagonal elements indicate misclassifications. Here, the diagonal shows really good results and the numbers off the diagonal are really small, confirming the good performance of the model.

Table 5: Confusion Matrix for the Best Model

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
0	39	0	0	0	3	0	0	0	0	0	0	0	0	1	0
1	0	48	0	0	0	0	0	0	0	0	0	0	0	0	0
2	2	0	57	2	0	0	0	0	0	1	0	0	0	0	0
3	0	0	1	41	0	0	0	0	0	0	0	0	0	0	0
4	3	0	0	1	52	0	0	0	0	0	0	0	0	2	0
5	0	0	0	0	0	68	1	0	0	0	3	0	0	0	0
6	0	0	0	0	0	0	66	0	0	0	0	0	0	0	0
7	0	0	0	0	0	1	0	47	0	0	2	1	1	0	0
8	0	0	0	0	0	0	0	0	59	0	0	2	1	0	0
9	0	0	0	0	0	0	5	1	0	66	3	0	0	0	0
10	0	0	0	0	0	5	0	2	0	4	71	0	0	0	0
11	0	0	0	0	0	0	0	1	0	0	0	57	0	0	0
12	0	0	1	0	0	0	1	0	3	0	0	1	65	0	0
13	3	0	0	1	0	0	0	0	0	0	0	0	0	39	0
14	1	0	4	0	0	0	0	0	0	0	0	0	0	1	57

Gaussian Blurring

The model was trained to classify the kernel size of a Gaussian blur applied to input images. The training and validation performance showed rapid initial learning, with the model achieving high accuracy early in the process.

Gaussian Blurring Pretext Task

The model starts with a train loss of 0.3049 and a validation loss of 0.0762, achieving a validation accuracy of 97.43%. This indicates a really good initial fit. In later epochs, the model continues to perform well, with validation accuracy mostly above 97%, showing that the model quickly learns to classify the blurred images correctly. The highest validation accuracy is 99.60% at epoch 14, indicating that the model is highly effective at classifying the kernel size of Gaussian blur applied to the images. We see some flunctuations in the validation, which is not that bad, but maybe it suggests occasional overfitting or difficulty in generalizing at certain points. The drop in accuracy at epoch 16 suggests potential overfitting or an outlier in the data.

Table 6: Training and Validation Performance for Gaussian Blurring Pretext Task

Epoch	**Train Loss**	**Validation Loss**	**Validation Accuracy (%)**
1	0.3049	0.0762	97.43
2	0.1271	0.0207	99.09
3	0.0801	0.1238	95.05
4	0.0756	0.0350	98.77
5	0.0713	0.0568	97.99
6	0.0462	0.0271	99.04
7	0.0478	0.0136	99.51
8	0.0253	0.1220	95.74
9	0.0784	0.0384	98.48
10	0.0625	0.0645	98.06
11	0.0506	0.1889	91.70
12	0.0238	0.0237	99.29
13	0.0246	0.1163	95.58
14	0.0663	0.0155	**99.60**
15	0.0224	0.0423	98.48
16	0.0485	0.7074	76.28

Scene classification using pretext Blurring

Following the pretext task, the model was fine-tuned for the main task of scene classification. In this step, the Gaussian kernel classifier part was replaced with the original classifier, and only the classifier part was fine-tuned while the feature extraction part was frozen. The results for the scene classification task are less impressive, with a validation accuracy improving modestly from 31.33% to 42.36% over 16 epochs. This performance suggests that the features learned during the Gaussian blur pretext task did not transfer effectively to the scene classification task.

Table 7: Training and Validation Performance for Scene Classification Using Pretext Task

Epoch	Train Loss	Validation Loss	Validation Accuracy (%)
1	2.2534	2.1095	31.33
2	2.1094	2.0847	34.34
3	2.0400	1.9966	37.35
4	1.9984	2.0066	38.80
5	1.9587	1.9520	37.90
6	1.9640	1.9225	39.58
7	1.9518	1.9146	39.35
8	1.9610	1.9737	39.02
9	1.9667	1.9260	38.80
10	1.9239	1.8989	39.69
11	1.9142	1.9087	40.91
12	1.8960	1.8867	40.91
13	1.8908	1.8777	41.47
14	1.8862	1.8390	**42.36**
15	1.8638	1.8646	40.91
16	1.8606	1.8817	40.25

Test evaluation

The test performance for scene classification confirmed these findings. The detailed class performance analysis showed varied results across different classes, with high precision and recall for some classes and poor performance for others. This discrepancy suggests that the pretext task may not have provided features that are highly relevant to the downstream task. The Gaussian Blurring pretext task is primarily focused on texture and edge detection, which are lower-level features. These features are useful but not entirely sufficient for the high-level semantic understanding required for scene classification. Scene classification often relies on understanding complex relationships and contextual information within the image, which might not be well captured by a task focused on blur detection.

While the EfficientNet-B0 model pretrained on ImageNet provides a robust starting point, the transfer of learned features from a task like Gaussian blur classification to scene classification might not be straightforward. The features learned during the pretext task might not align well with the features needed for differentiating between scenes like kitchens, streets, forests, etc. The model has learned to recognize general object categories but not specific scenes.

The 15-Scene dataset contains diverse and complex scenes which require high-level contextual understanding. Features learned from Gaussian blurring might not directly translate to distinguishing these complex scenes.

The gradual improvement in validation accuracy from 31.33% to 42.36% and the test accuracy of 40.25% suggest that the model is learning but at a slower pace. This is typical when the pretext task does not provide highly transferable features. Additionally, the fluctuations in loss and accuracy indicate that the model is still adapting to the new task and may need more epochs or better optimization strategies. That also depends on the number of epochs, so we don't overfit the model, but in later epochs it may have been some improvement.

Table 8: Precision, Recall, and F1-Score for Blurring

Precision	Recall	F1-Score
0.11	0.02	0.04
0.49	0.71	0.58
0.18	0.13	0.15
0.25	0.10	0.14
0.18	0.14	0.16
0.54	0.36	0.43
0.47	0.79	0.59
0.52	0.67	0.59
0.42	0.18	0.25
0.32	0.25	0.28
0.40	0.54	0.46
0.41	0.40	0.40
0.44	0.37	0.40
0.57	0.67	0.62
0.32	0.65	0.43

Table 9: Confusion Matrix for Scene Classification Test for Blurring

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	4	6	0	9	2	0	2	1	2	1	0	4	8	3
1	34	1	0	4	0	0	1	0	0	0	0	0	0	7
1	2	8	3	1	1	2	4	1	5	8	4	6	0	16
2	4	4	4	4	0	1	0	0	1	0	1	1	4	16
1	9	8	5	8	1	0	3	1	2	0	1	4	3	12
0	0	2	1	3	26	3	9	1	6	14	1	2	3	1
0	1	0	0	1	0	52	0	0	0	5	1	3	0	3
0	0	0	0	0	7	0	35	0	0	2	5	1	1	1
0	2	2	3	0	1	6	0	11	6	4	10	7	0	10
0	1	5	0	0	1	12	4	0	19	23	4	2	0	4
0	0	1	0	2	6	14	4	3	4	44	0	2	0	2
0	3	2	0	0	0	4	3	5	8	3	23	1	0	6
0	2	3	0	1	3	12	2	3	2	7	3	26	3	4
2	2	0	0	9	0	0	0	0	0	0	0	0	29	1
1	5	2	0	2	0	5	0	0	4	0	3	0	0	41

Black and White Perturbation

The model was trained to classify images that have undergone black and white perturbation. The training and validation performance showed consistent learning, with the model achieving high accuracy early in the process.

Perturbation Pretext Task

The model starts with a train loss of 0.2763 and a validation loss of 0.1920, achieving a validation accuracy of 87.51%. This indicates a reasonable initial fit. As the epochs increase, validation accuracy stabilizes around 88%, with some fluctuations in validation loss, indicating the model's continued efforts to generalize. The model achieves its highest validation accuracy of 89.57% at epoch 3, with a validation loss of 0.1721, indicating effective learning early in the training process.

Table 10: Training and Validation Performance

Epoch	Train Loss	Validation Loss	Validation Accuracy (%)
1	0.2763	0.1920	87.51
2	0.1960	0.2412	86.00
3	0.1872	0.1721	**89.57**
4	0.1864	0.1758	87.90
5	0.2158	0.1880	87.28
6	0.1781	0.1791	89.07
7	0.1756	0.1742	88.73
8	0.1622	0.1660	88.29
9	0.1579	0.1633	88.57
10	0.1609	0.1573	89.18
11	0.1604	0.1731	87.79
12	0.1614	0.1614	88.73
13	0.1845	0.1736	88.85
14	0.1840	0.1646	88.85
15	0.1685	0.1634	87.34
16	0.1696	0.1623	88.29

Scene classification using perturbation

The results for the scene classification task following the perturbation pretext task indicate that the model struggled to learn effectively. The train loss starts at 2.7177 and fluctuates slightly throughout the epochs but remains high, suggesting that the model is not effectively minimizing the loss. Similar to the train loss, the validation loss starts high at 2.6915 and shows minor fluctuations but does not decrease significantly, indicating poor generalization. The best validation accuracy achieved is 12.26% at epoch 13, which is insufficient for a reliable classification model.

Table 11: Training and Validation Performance for Scene Classification Using Perturbation Pretext Task

Epoch	Train Loss	Validation Loss	Validation Accuracy (%)
1	2.7177	2.6915	9.03
2	2.7055	2.6590	10.81
3	2.6710	2.7420	11.37
4	2.6913	2.6941	9.36
5	2.6723	2.6692	11.82
6	2.6688	2.6753	9.92
7	2.6695	2.6888	11.59
8	2.6765	2.6851	10.48
9	2.6787	2.6370	11.59
10	2.6630	2.6513	10.59
11	2.6485	2.6594	12.15
12	2.6531	2.7020	10.70
13	2.6703	2.6615	**12.26**
14	2.6634	2.6550	11.26
15	2.6510	2.6438	11.15
16	2.6655	2.6769	10.26

Test evaluation

The test performance for scene classification confirmed these findings, as well as the previous task with blurring. The performance metrics across all classes show very low precision, recall, and F1-scores, indicating that the model is not performing well in predicting any class effectively. The perturbation pretext task focused on distinguishing between perturbed and non-perturbed images, which may not have provided the model with features that are highly relevant for complex scene classification. Scene classification requires understanding higher-level contextual features that were not emphasized in the pretext task.

While pretext tasks can help in learning useful representations, the effectiveness of transfer learning heavily depends on how relevant the pretext task is to the downstream task. The features learned for perturbation detection likely did not generalize well to the intricate and diverse features needed for scene classification.

Same as in the blurring task, EfficientNet-B0 is optimized for object recognition. The fine-tuning done after the perturbation task was insufficient to adapt the model for scene classification. The feature extraction layers, possibly frozen during fine-tuning, limited the model's ability to adapt to the new task. The fine-tuning process may not have been extensive enough. Typically, more epochs, a different learning rate, or unfreezing some of the feature extraction layers can help in better transferring the learned features, but can cause overfitting in the future.

The results below can confirm this performance.

Table 12: Test results for Perturbation task

Class	Precision	Recall	F1-Score
0	0.00	0.00	0.00
1	0.00	0.00	0.00
2	0.05	0.02	0.02
3	0.00	0.00	0.00
4	0.00	0.00	0.00
5	0.12	0.56	0.20
6	0.11	0.23	0.15
7	0.00	0.00	0.00
8	0.10	0.03	0.05
9	0.07	0.13	0.09
10	0.08	0.18	0.11
11	0.16	0.14	0.15
12	0.07	0.01	0.02
13	0.00	0.00	0.00
14	0.00	0.00	0.00

Table 13: Confusion Matrix for Scene Classification for perturbation task

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0	3	0	0	15	5	0	0	10	10	0	0	0	0
0	0	1	0	0	27	3	0	0	9	6	2	0	0	0
0	0	1	0	0	19	11	0	4	8	11	5	3	0	0
0	0	0	0	0	10	7	0	0	10	9	6	0	0	0
0	0	1	0	0	27	8	0	0	5	17	0	0	0	0
0	0	1	0	0	40	4	0	1	11	11	2	2	0	0
0	0	4	0	0	16	15	0	4	6	15	5	0	0	1
0	0	2	0	0	28	7	0	0	6	8	0	1	0	0
0	0	3	0	0	5	14	0	2	15	16	4	3	0	0
0	0	1	0	0	29	10	0	1	10	19	5	0	0	0
0	0	0	0	0	39	9	0	0	12	15	5	2	0	0
0	0	2	0	0	13	8	0	2	5	19	8	1	0	0
0	0	0	1	0	20	15	2	1	9	20	2	1	0	0
0	0	0	0	0	23	4	0	0	13	2	1	0	0	0
0	0	3	0	0	12	14	2	6	12	8	5	1	0	0

Perturbation pretext task examples performance

In this section I realized that when I rerun the code for the 3 epochs, I get different results than the time I run it with 16 epochs. My thought is that there is something wrong with the splitting of the data, when I run it twice, even though every time I run it with the same epoch, I get the same results. So, my comparison will be analogic and not with the exact numbers, because I find it weird to run the same code again and have different results for different epochs. I would expect to have the same results, and compare the evolution of the performance of the model in the different epochs.

The accuracy I get from the run of 16 epochs is 10.26% in the 16th epoch, which is less than the highest accuracy that it occured in 13 epochs. That may show that after some epochs the model overfit. The accuracy for 3 epochs is 23.07% in the 3rd epoch, while it improved from 20.06% that it was in the first epoch. That shows an improvement, that may mean that 3 epochs is not enough and the model might have better performance in later epochs. In general, the perturbation task may not align well with the requirements of scene classification. Training for many epochs on the perturbation task might not improve, and could even degrade the model's ability to transfer useful features. But for not that many epochs, we may not have enough information about it and lose performance that could have been better in some later epochs.

Comparison of performances of the models

From the 3 models, the fully-supervised model outperforms both self-supervised models significantly, achieving an accuracy of 92.75%. This high accuracy indicates that training directly on the scene classification task with labeled data provides the best performance.

The model trained using the Gaussian blurring pretext task achieves a moderate accuracy of 42.36%, which is considerably higher than random guessing but still far below the fully-supervised model. This indicates that while the Gaussian blurring task helps in learning some useful features, it is not sufficiently aligned with the scene classification task to achieve high performance. The confusion matrix and detailed metrics show that some classes are better classified than others, suggesting that the pretext task did help in learning some discriminative features.

The model trained using the perturbation pretext task achieves a very low accuracy of 12.26%. This performance is only marginally better than random guessing, indicating that the features learned during the perturbation task are not useful for scene classification. The detailed metrics and confusion matrix show poor performance across almost all classes, highlighting the ineffectiveness of this pretext task for the downstream scene classification task.

Table 14: Comparison of Accuracies for Different Models

Model	**Accuracy (%)**
Fully-Supervised Model	92.75
Self-Supervised Model (Gaussian Blurring Pretext Task)	42.36
Self-Supervised Model (Perturbation Pretext Task)	12.26

The fully-supervised model is directly trained on the scene classification task, ensuring that all learned features are relevant for this specific task. This direct approach leads to much higher performance. The Gaussian blurring pretext task, while somewhat relevant, primarily focuses on low-level features like texture and edges, which are not sufficient for high-level scene understanding required for scene classification. The perturbation pretext task focuses on detecting image perturbations, which do not provide meaningful features for understanding and classifying complex scenes.

So, the self-supervised scheme was not particularly useful for classifying the scene classes in this context. While the model trained using the Gaussian blurring pretext task achieved a moderate accuracy of 42.36%, it still fell significantly short of the fully-supervised model's accuracy of 92.75%. The perturbation pretext task performed even worse, with an accuracy of only 12.26%. These results suggest that the features learned during the self-supervised tasks did not transfer effectively to the downstream scene classification task. The fully-supervised approach, which directly trained the model on the scene classification data, resulted in much higher accuracy and better overall performance, highlighting the importance of task relevance in feature learning.

Strategies for underfitting and overfitting

1. Various combinations of learning rates and batch sizes were tested to find the optimal settings that would prevent underfitting and overfitting. Smaller learning rates were chosen to ensure gradual convergence, and different batch sizes were used to balance between training speed and stability. The model was trained and validated using different combinations, and the best performing model was selected based on the best validation accuracy.

2. Dropout was applied to the classifier part of the EfficientNet-B0 model to reduce overfitting by randomly dropping units during training. This technique helps in preventing the model from becoming too reliant on specific neurons and improves its ability to generalize.

3. A consistent random seed was set using the set_seed function to ensure reproducibility of the results and to control the randomness in data shuffling, weight initialization, and other stochastic processes. This helps in maintaining consistent training behavior across different runs.

The training and validation loss plot was presented to show if there is overfitting or underfitting and we see that the model converges well without significant overfitting. The confusion matrix shows high precision and recall across most classes, indicating that the model is correctly identifying the majority of the images and generalizing well to the test data. Misclassifications are minimal, suggesting that the model is not overfitting to the training data.