Model Evaluation

EfficientNT:	Accuracy	Precision	Recall	F1-score	GFLOPs
90:10	0.57	0.57	1.00	0.73	0.8031
80:20	0.58	0.58	1.00	0.73	0.8031
70:30	0.58	0.58	1.00	0.73	0.8031
60:40	0.58	0.58	1.00	0.73	0.8031
50:50	0.58	0.58	1.00	0.73	0.8031
40:60	0.58	0.58	1.00	0.73	0.8031
30:70	0.58	0.58	1.00	0.73	0.8031
20:80	0.58	0.58	1.00	0.73	0.8031
10:90	0.58	0.58	1.00	0.73	0.8031

ResNeT 50	Accuracy	Precision	Recall	F1-score	GFLOPs
90:10	0.57	0.57	1.00	0.73	7.7553
80:20	0.58	0.58	1.00	0.73	7.7553
70:30	0.58	0.58	1.00	0.73	7.7553
60:40	0.58	0.58	1.00	0.73	7.7553
50:50	0.58	0.58	1.00	0.73	7.7553
40:60	0.58	0.58	1.00	0.73	7.7553
30:70	0.58	0.58	1.00	0.73	7.7553
20:80	0.58	0.58	1.00	0.73	7.7553
10:90	0.58	0.58	1.00	0.73	7.7553

ResNeT101	**Accuracy**	**Precision**	**Recall**	**F1-score**	**GFLOPs**
90:10	57	47	57	43	15.1987
80:20	58	59	58	44	15.897
70:30	58	65	58	44	15.1987
60:40	59	68	51	40	15.1987
50:50	58	76	58	42	15.1987
40:60	58	76	58	42	15.1978
30:70	58	76	58	42	15.1987
20:80	58	76	58	42	15.1987
10:90	58	33	58	42	15.1987

googleNet	Accuracy	Precision	Recall	F1-score	GFLOPs
90:10	100	100	100	100	1.50
80:20	99.22	99.10	99.33	99.21	1.50
70:30	98.45	98.49	98.33	98.41	1.50
60:40	98.83	98.94	98.68	98.80	1.50
50:50	97.36	97.42	97.18	97.29	1.50
40:60	95.73	96.40	95.05	95.57	1.50
30:70	95.34	96.12	94.57	95.16	1.50
20:80	92.23	93.89	90.91	91.82	1.50
10:90	91.03	93.02	89.51	90.50	1.50

MobileNetV2	Accuracy	Precision	Recall	F1-score	GFLOPs
90:10	0.93	0.93	0.93	0.93	0.6153
80:20	0.93	0.93	0.93	0.93	0.6153
70:30	0.92	0.92	0.92	0.92	0.6153
60:40	0.89	0.89	0.89	0.89	0.6153
50:50	0.91	0.91	0.91	0.91	0.6153
40:60	0.89	0.89	0.89	0.89	0.6153
30:70	0.89	0.89	0.89	0.89	0.6153
20:80	0.89	0.89	0.89	0.89	0.6153
10:90	0.79	0.79	0.79	0.79	0.6153

Discussion:

The experimental results clearly indicate that **GoogleNet** outperformed all other evaluated models across every data split. With **accuracy, precision, recall, and F1-scores consistently ranging between 91% and 100%**, GoogleNet demonstrated remarkable stability and robustness. This superior performance can be attributed to its **Inception architecture**, which utilizes multi-scale convolutional kernels (1×1, 3×3, and 5×5) within the same layer. This design allows GoogleNet to extract both fine-grained and high-level features simultaneously, enhancing its ability to learn complex visual representations. Furthermore, despite its high accuracy, the model remains computationally efficient with only **1.50 GFLOPs**, indicating an excellent trade-off between performance and resource usage. This efficiency makes it suitable for real-time or resource-constrained environments.

MobileNetV2 also achieved commendable results, maintaining an accuracy of approximately **0.89–0.93** with the lowest computational cost among all models (**0.6153 GFLOPs**). Its lightweight structure, built upon **depthwise separable convolutions** and **inverted residuals**, enables efficient feature extraction with minimal parameter overhead. However, its slightly reduced accuracy compared to GoogleNet suggests that its compact architecture may limit the model's capacity to capture highly abstract or complex features, particularly when the dataset contains diverse or fine-grained classes.

In contrast, **EfficientNT**, **ResNet50**, and **ResNet101** showed significantly weaker performance, with accuracy values stagnating around **0.57–0.58** across all data splits. The consistently poor results indicate that these models struggled to converge effectively or generalize to unseen data. Several factors likely contributed to this outcome:

1. **Over-parameterization and overfitting:**
   ResNet50 and ResNet101 are deep architectures with high computational requirements (**7.75 and 15.19 GFLOPs**, respectively). When applied to smaller or less diverse datasets, their large number of parameters can lead to overfitting, where the model memorizes training patterns rather than learning generalizable features.

2. **Suboptimal training or hyperparameter tuning:**
   Deep models such as ResNet often require careful tuning of learning rates, batch normalization, and regularization techniques. Without proper optimization, they may fail to exploit their full representational potential.

3. **Inefficient feature utilization:**
   EfficientNT (a likely lightweight custom network) achieved only **0.57–0.58 accuracy**, suggesting that its architecture might not have been sufficiently deep or expressive to capture the complexity of the target data. Despite having moderate computational cost (**0.8031 GFLOPs**), the network's limited representational capacity likely restricted its performance.

Overall, **GoogleNet** proved to be the most effective model in balancing **accuracy, computational efficiency, and generalization capability**. It achieved consistently superior results across all train-test ratios, highlighting its strong adaptability to varying data distributions. **MobileNetV2** followed as a viable alternative for scenarios prioritizing low computational cost over maximum accuracy. Conversely, the **ResNet** models and **EfficientNT** failed primarily due to a mismatch between their architectural complexity and the dataset characteristics, underscoring the importance of selecting architectures that are both computationally appropriate and well-optimized for the given task.