Journal of Computers and Intelligent Systems
Vol. 3, No. 2, 2025, pages 121 - 138
Journal Homepage:
https://journals.iub.edu.pk/index.php/JCIS/

Evaluating GoogleNet's Image Classification Performance: Impact of Dataset Size, Balancing, and Splits Ratios

Mariyam Amreen[1], Mujeeb-ur-Rehman[2]
[1,2]Department of Computer Science, University of Management and Technology, Sialkot, Pakistan

ARTICLE INFO

Article History:

Received:	April	12, 2025
Revised:	June	25, 2025
Accepted:	June	26, 2025
Available Online:	June	27, 2025

Keywords:

Image Classification
Dataset Splitting
Computer Vision
GoogleNet
Deep Learning

Classification Codes:

Funding:

This research received no specific grant from any funding agency in the public or not-for-profit sector.

ABSTRACT

Image classification is one of the fields of utmost importance in computer vision that has numerous applications in real-world scenarios. GoogleNet is one of the commonly used deep models that is especially utilized for object detection and image classification by learning and understanding visual patterns. This study investigates how the GoogleNet model performs on image classification using three factors: Sample size, dataset balance, and various train-test split ratios. Model accuracy was tested on the CIFAR-10 dataset by trying it out on dataset sizes of 25%, 50%, 75%, and 100%, both with and without balance. The results show that both the size and balance of the dataset have a direct impact on classification accuracy, with balanced datasets always yielding higher accuracy rates compared to unbalanced datasets. In addition, when comparing various train-test ratios 50%-50%, 60%-40%, 70%-30%, 80%-20%, and 90%-10% the best performance of the model was achieved when it was trained on 70% of the data and tested against the other 30%.

Corresponding Author's Email: mujeeb.rehman@skt.umt.edu.pk

1. Introduction

Image classification is a critical task in computer vision with implications in a vast number of domains ranging from healthcare [1], through agriculture [2], to e-commerce [3]. Standard machine learning (ML) methods typically rely on manually-designed features, which do not generalize well across different datasets. However, advancements in deep learning (DL) notably by Convolutional Neural Networks (CNNs) significantly enhanced image classification applications by simplifying feature extraction and model generalization across disparate kinds of data. Despite such advances, execution of high-performance CNNs on low-computational devices remains a challenge. To address this issue, GoogleNet [4] was introduced, introducing a structure that minimized computational and memory utilization through tools such as inception modules and depthwise separable convolutions with competitive levels of accuracy maintained. Though GoogleNet is more famous for being highly efficient, its performance is based on

several factors in total, which vary from hyperparameter setting, data size, class balance, and proportion of the data allocated for training and testing. Hyperparameters like learning rate, batch size, and number of training epochs form the core of controlling how well and accurately the model learns from data [5]. The poor settings can result in problems such as slow convergence, underfitting, or overfitting. Another major issue is class imbalance in data, which results in predictions made by the model biased towards majority classes and thus reducing its effectiveness in classifying minority instances [6]. Various strategies such as oversampling, undersampling, and synthetic data generation approaches like SMOTE [7] have been tried to combat this issue. Additionally, dataset size is a critical parameter in quantifying a model's ability to generalize well [8]. The train-test split ratio used to split data into training and test sets also directly affects model performance [9], since a bad split might leave the model with insufficient training data or provide unstable evaluation. While individual effects of dataset size [8], class balancing [10], and train-test split ratios [9] have been studied in earlier studies, there are not many studies on how these factors affect performances of models like GoogleNet as a whole. This research aims to fill this knowledge gap by exploring the interaction of these factors on GoogleNet's image classification performance on the CIFAR-10 dataset. The objectives of this study are: (1) to evaluate the effect of varying dataset sizes, (2) to examine the influence of the impact of class balance on the outcome of classification, and (3) to find the best train-test split ratio for enhancing the model generalization.This study makes several important contributions. Its primary contributions include a thorough evaluation of GoogleNet's performance across different sizes of data sets, an examination of how balanced and imbalanced data sets impact model accuracy, and a comparative analysis of the effects of various train-test split ratios on generalization performance.The remainder of this paper is organized as follows: Section 2 provides a literature review, Section 3 provides the methodology adopted, Section 4 provides the experimental setup, Section 5 provides the results, and Section 6 concludes the research with the findings and the recommendations for future research.

2. Related Work

GoogLeNet has gained popularity because of its efficient architecture, making it suitable for real-time applications and lowering the cost of computations compared to previous deep learning architectures. Szegedy et al., [11] introduced GoogLeNet optimized for computational cost by lowering parameters and memory consumption through Inception modules and global average pooling. Subsequent architectures extended this further through the inclusion of depthwise separable convolutions and other optimizations to make it more efficient while being competitive on accuracy. Despite its popularity, there has been limited research on how differences in train-test split ratios, class balance, and dataset size influence GoogLeNet's classification accuracy. The significance of dataset size was already noted in earlier studies of deep learning. The study by Chen Sun et al. [12] shows that larger datasets increase generalization but that the increment decreases after a certain point. Class variance is another significant challenge in supervised learning. Models trained on imbalanced datasets tend to acquire bias toward majority classes, and the minority class instances are not well identified. Research by Nitesh V. Chawla et al. [7] highlights how imbalanced data would impact model predictions in a negative direction. Mateusz Buda et al. [14] studied the impact of class imbalance on CNNs and compared various methods to combat this issue.

Research conducted by Spelmen Vimalraj [15] suggests that the SMOTE technique is an effective method to address class imbalance but other techniques such as oversampling and undersampling have also been examined since they can be utilized as solutions [16]. Ratios of train-test splits are the most important factor in evaluating the model. Houda Bichri et al. [9] tested different split ratios, i.e., 60%-40%, 70%-30%, 80%-20%, and 90%-10%, to study their influence on the executions of a number of pre-trained models. Further, Ismail Olaniyi Muraina [17] studied the effect of different training-testing splits variations on model performance and concluded that the optimal ratio is dataset size-dependent, which have direct impacts on generalization. Although these studies offer convenient insights into individual factors that influence model performance, not much research systematically examines their overall impact.

Recent studies have focused on refining CNN-based models for medical, agricultural, and IoT applications, making their performance in real-world applications even higher. Unnisa et al. [18] studied the effect of hyperparameter tuning on convolutional neural networks for detecting skin cancer, and found that parameter tunning significantly increases classification performance in complex medical imaging tasks. In parallel, Sarwar et al. [20] combined few-shot learning with transfer learning on breast cancer detection, which effectively mitigated the difficulties of small dataset and improved the efficiency of models.

In the context of computation overhead efficiency, Shah et al. [19] proposed an efficient and lightweight signcryption scheme for secure communication in UWSNsFor secure available resources optimization, we can observe that the timely demand of optimized resource-aware architectures like GoogLeNet in resource constrained environment cannot be overemphasised. Furthermore, in the domain of predictive analytics, Ayub et al. [21] proposed a multi-level deep learning autoencoder model for parametric time series forecast, which demonstrated useful versatility of CNN-based paradigms beyond typical classification problems.

CNN architectures have also been used in the field of agriculture and environmental monitoring. Wang et al. [23] developed a hybrid deep learning method for early rice barn shade-disease detection within polytunnel IoT-based smart agriculture which reflects how adapted CNN models can be used in low resource, real world scenarios.Ullah et al. [24] proposed a machine learning-based intelligent decision-making system for energy-efficient fog node selection and intelligent switching in IoT networks, demonstrating the real-world applicability of lightweight yet high-accuracy models for distributed systems.
In medical image processing, Sarwar et al. [22] proposed the combination of deep learning and ant colony optimization to enable accurate segmentation of skin lesions, with notable improvement in detection accuracy and computational cost. Similarly, Akram et al. [25] proposed an edge-weighted texture feature extraction technique for breast cancer diagnosis from histopathological images, as an alternative to CNN-based models for image classification problems that are challenging to solve. This research will fill that gap by assessing the accuracy of GoogLeNet classification for the CIFAR-10 dataset with different dataset sizes, class balancing methods, and train-test split ratios. The results would be very valuable to guide researchers and practitioners in enhancing deep learning models for image classification.

3. Used Approach

The research process utilized in this research is illustrated in Figure 1. It starts with gathering the dataset needed to classify images based on the GoogleNet model. Once the data is collected, it is separated into four portions 25%, 50%, 75%, and 100% to analyze the performance of the model based on varying dataset sizes. Each segment is also analyzed under two scenarios: balanced and unbalanced, to measure the impact of class distribution. To further investigate the impact of train-test data ratios, the dataset is divided into five different splits: 50%-50%, 60%-40%, 70%-30%, 80%-20%, and 90%-10%. Image preprocessing is done through the Keras library with operations like resizing, rescaling, and application of data augmentation processes such as shear transformations, zooming, and horizontal flipping. The model is configured with the best hyperparameters, which include the Adam optimizer, a learning rate of 0.001, and categorical cross-entropy as the loss. Training is conducted over 10 epochs with a batch size of 32. For the purpose of performance analysis, a number of metrics are taken into account, such as accuracy, precision, recall, F1-score, and ROC-AUC, in addition to computational measures like training time, memory consumption, and time complexity. All these metrics provide a comprehensive evaluation of the model's effectiveness and efficiency in performing image classification tasks.

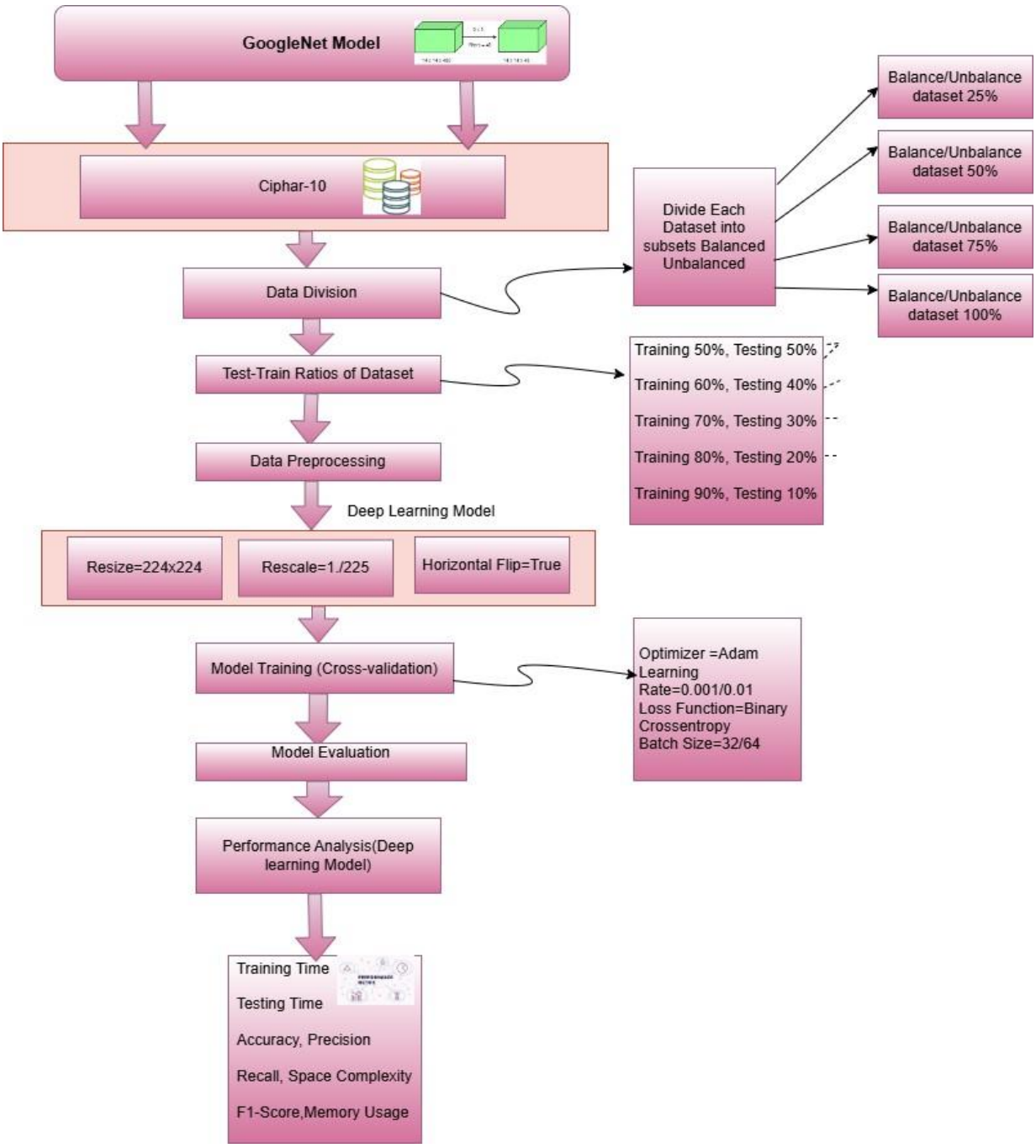

Figure 1.Proposed Methodology Workflow for GoogleNet Image Classification Experiments

The hyperparameters we used in our experimentation are shown in Table 1 along with their purpose and values.

Table 1.List of Hyper-Parameters along with their purpose and values used in the experimentation of Googlenet

Hyperparameter	**Purpose**	**Value**
Optimizer	Algorithm used for optimization.	Adam
Learning Rate	Learning Rate control weight update size	0.001
Loss Function	Loss Function measures prediction Error	Binary Cross- Entropy
Data	Enhances dataset diversity through	Rescale=255, Shear=0.2, Zoom=0.2,

Augmentation	transformations.	Flip=True
Activation Function	Introduces non-linearity for com- plex pattern learning.	ReLU (Conv and FC layers)
Epochs	Epochs define data pass through the model	10
Batch Size	Number of samples processed in a batch.	32
Stride	Stride sets filter moment in pooling	1
Padding	Padding preserves image dimension	Same Padding
Fully Connected Layer Size	Fully Connected layers define neurons before output	128
Momentum	Momentum smooths training updates	Not applicable (Adam used)
Weight Decay	Regularization parameter to pre- vent overfitting.	0.0001
Dropout Rate	Fraction of neurons dropped during training.	0.5

4. Experimentation

The data used in this research were obtained from Kaggle, and the testing was done on both the local hardware and cloud utilizing Google Colab. The local system used to deploy models was Windows 11 Home with an Intel(r) Celeron(r) N4120 processor at 1.10 GHz, and 4.00 GB of RAM (3.82 GB available). Such hardware setups had some computation constraints, particularly when working with computationally intensive deep learning processes. To avoid these constraints, the T4 GPU on Google Colab was utilized, which provided a cloud environment of 15.0 GB RAM and approximately 50 GB of available disk space. The environment supported increased model training and testing in deep learning processes. Python was employed as the underlying programming language for experiments due to its extensive libraries and frameworks supporting model construction, training, and testing.

For the optimization, Adam optimizer was employed due to its empirical stability and adaptability in CNN-based image classification tasks, as suggested by Wojciuk et al. [5]. The learning rate was set to 0.001, striking a balance in providing training stability and having an acceptable convergence rate. Training was performed with a batch size of 32 for 10 epochs, a configuration reached through preliminary experimentation for providing stable performance without overfitting within available computing resources. Data augmentation methods, such as shear transformations, zooms, and flips, were used to enhance the diversity of the dataset and enhance the model's capacity for generalization. The augmentation methods were selected specifically because they introduce natural variations in the data without warping the intrinsic properties of an image, a procedure that is classically demonstrated by Szegedy et al. [12] and other related CNN-based research.

This study used the CIFAR-10 dataset, as originally suggested by Krizhevsky et al., and widely used for benchmarking image classification models. Downloaded from the Kaggle website, the CIFAR-10 dataset consists of 60,000 color images with dimensions 32×32 pixels and weighs around 163.2 MB. It is split into three broad categories: training, test, and validation. It contains 50,000 images in the training set and 10,000 in the test set. Every image is assigned to one of ten pre-defined classes: airplane, automobile, bird, cat, deer, dog,

frog, horse, ship, and truck. Although some datasets keep images in separate folders per class, CIFAR-10 keeps data in a bundled format, and therefore label mapping is needed in pre-processing. One of the primary advantages of CIFAR-10 is that it provides class distribution balance with 5,000 training images and 1,000 testing images per class, enabling fair training of the model and unbiased performance measurement.

To estimate the effect of dataset size and class distribution on model effectiveness, the test was run on four proportionally increasing segments of the dataset: 25%, 50%, 75%, and 100% of the CIFAR-10 dataset. Two instances were prepared for each proportion, one with balanced distribution over all classes and another that introduced class imbalance. This two-stage approach enabled an exhaustive exploration of how both dataset size and distribution impact classification accuracy. In the balanced subsets of the dataset, both classes were represented equally to enable equal training and testing. The unbalanced subsets, on the other hand, had more images in the majority classes than in the minority classes. In the 25% subset, for example, there were 2,000 images per majority class and 1,000 per minority class, totaling 15,000 images. There were 30,000 images in the 50% subset, with 4,000 per majority class and 2,000 per minority class. There were 6,000 and 3,000 images for majority and minority classes respectively in the 75% subset, totaling 45,000 images. Finally, the full 100% subset consisted of 60,000 images, with 8,000 per majority class and 4,000 per minority class.

The detailed specifications of Balanced and Unbalanced datasets at various proportions are provided in Table 2.

Table 2.Specification of Balanced and UnBalanced Ciphar-10 Dataset Across Different division

Dataset Type	Size	Total Samples	Images per Class	Dominant (5)	Minor (5)
25% Balanced	25%	15,000	1,500	1,500	1,500
50% Balanced	50%	30,000	3,000	3,000	3,000
75% Balanced	75%	45,000	4,500	4,500	4,500
100% Balanced	100%	60,000	6,000	6,000	6,000
25% Unbalanced	25%	15,000	Dom: 2,000	2,000	1,000
50% Unbalanced	50%	30,000	Dom: 4,000	4,000	2,000
75% Unbalanced	75%	45,000	Dom: 6,000	6,000	3,000
100% Unbalanced	100%	60,000	Dom: 8,000	8,000	4,000

5. Result & Analysis

The Efficiency of the GoogleNet model on 25%, 50%, 75%, and 100% balanced and unbalanced datasets are presented in Table 3.

Table 3.Evaluation Metrices of Balanced and Unbalanced Datasets Across Different Divisons

Ratio	Accuracy	Precision	Recall	F1-Score	ROC-AUC
25% Balanced	80.43%	0.6056	0.5743	0.5664	0.9167
50% Balanced	79.40%	0.7994	0.7940	0.7926	0.9725
75% Balanced	78.64%	0.7883	0.7864	0.7852	0.9756
100% Balanced	78.16%	0.7848	0.7816	0.7817	0.9759
25% Unbalanced	71.22%	0.7263	0.7122	0.7139	0.9597
50% Unbalanced	64.38%	0.6901	0.6438	0.6484	0.9365

75% Unbalanced	77.40%	0.7794	0.7740	0.7727	0.9738
100% Unbalanced	77.91%	0.7820	0.7791	0.7784	0.9732

For the 25% balanced dataset, the model performed with an accuracy of 80.43%, along with a precision of 0.6056, recall of 0.5743, F1-score of 0.5664, and a ROC-AUC score of 0.9167. On comparison, when the model was tested on the 25% unbalanced dataset, the performance decreased to an accuracy of 71.22%, precision of 0.7263, recall of 0.7122, F1-score of 0.7139, and a ROC-AUC score of 0.9597. Shifting to the 50% dataset size, balanced dataset gave 79.40% accuracy, precision of 0.7994, recall of 0.7940, F1-score of 0.7926, and a ROC-AUC score of 0.9725. In contrast, the unbalanced dataset had a significant drop in performance and achieved just 64.38% accuracy along with proportional declines in other measurement metrics, as shown in Table 3.

A particularly interesting observation was that the model's effectiveness on the 50% unbalanced dataset was worse than on the 25% unbalanced dataset. This can likely be attributed to the increased sampling imbalance as the dataset size grew without addressing class distribution. As the number of majority class samples expanded at 50%, the model became increasingly biased towards these dominant classes, neglecting minority class patterns. This overfitting to the majority classes negatively impacted overall classification accuracy. However, as the dataset size was further increased to 75% and 100%, performance improved progressively. This improvement is likely because the number of minority class samples also rose, giving the model better opportunities to learn from these underrepresented classes, which helped reduce overfitting and enhanced its generalization ability. Similar findings were reported by Buda et al. [14] and Unnisa et al. [18], who discussed how data imbalance affects CNN-based classification and how increasing minority class representation can mitigate its adverse effects.

At the 75% sample size, the balanced dataset continued to outperform the unbalanced one, with an accuracy of 78.64%, while the unbalanced dataset achieved 77.40%. Finally, when utilizing the complete 100% dataset, the balanced dataset recorded an accuracy of 78.16%, while the unbalanced dataset achieved a slightly lower accuracy of 77.91%. A consistent trend observed throughout the experiments was that increasing dataset size led to improvements in performance metrics for both balanced and unbalanced datasets. However, balanced datasets consistently delivered better results, particularly in terms of precision, recall, and F1-score. These trends are visually represented in Figure 2 and Figure 3, while resource usage details are provided in Table 4.

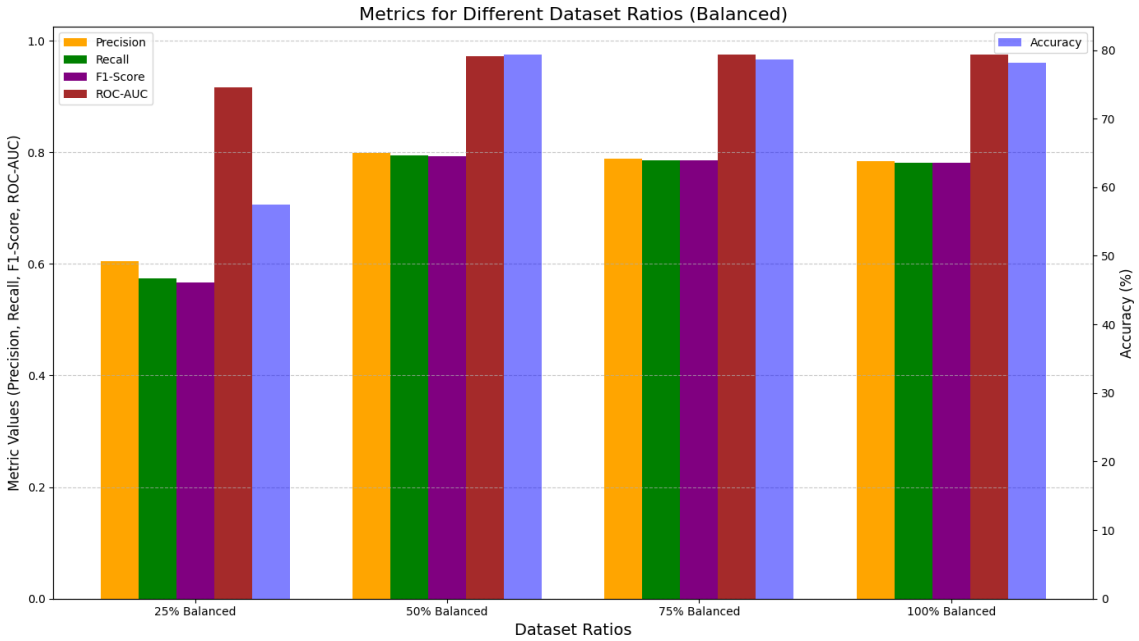

Figure 2.Performance Comparison of GoogleNet on Balanced CIFAR-10 Dataset at Different Dataset Sizes

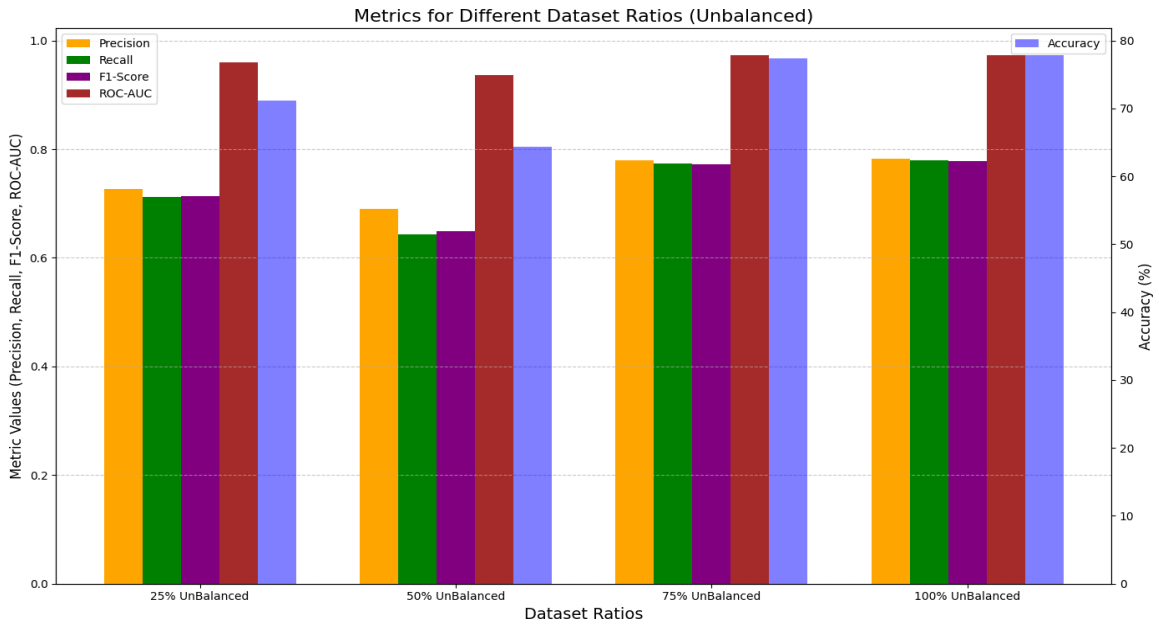

Figure 3.Performance Comparison of GoogleNet on Unbalanced CIFAR-10 Datasets at Different Dataset Sizes

The bar graph in Figure 2-3 consistent with earlier results as presented in Table 3. Resource usage metrics for evaluating the model's performance on balanced and unbalanced datasets at different proportions is

presented in Table 4

Table 4.Resource Usage Metrices of Balanced and Unbalanced Datasets Across Different Divisons

Ratio	Training	Testing	Time Complexity	Memory
25% Unbalanced	64.97 sec	4.90 sec	O(N*E*C)	1844.3 MB
50% Unbalanced	126.14 sec	5.78 sec	O(N*E*C)	1873.6 MB
75% Unbalanced	169.03 sec	4.91 sec	O(N*E*C)	1873.6 MB
100% Unbalanced	190.82 sec	4.75 sec	O(N*E*C)	1869.6 MB
25% Balanced	697.36 sec	2515.24 sec	47112729.52 Flops/sec	1464.8 MB
50% Balanced	210.02 sec	3.80 sec	673229.33 Flops/sec	1673.2 MB
75% Balanced	191.26 sec	5.05 sec	O(N*E*C)	2012.68 M
100% Balanced	212.68 sec	5.29 sec	O(N*E*C)	2042.3 MB

The performance measures attained by training and testing the model with the balanced and unbalanced datasets of four varied dataset sizes are shown in Table 4. Performance measures such as training time, testing time, total running time, memory space, and computational complexity were considered in this comparison. It is shown by the results that with the increase in the dataset size, the model takes more time to train and test, and it also needs more memory. Balanced datasets, in reality, are well known to require additional computation resources due to the balanced distribution of each class, thereby demanding more processing needs while training. For instance, with the 25% unbalanced dataset, the model trained in 64.97 seconds, tested in 4.90 seconds, and consumed 1844.30 MB of memory. The 25% balanced dataset consumed much more resources trained in 697.36 seconds, tested in 2515.24 seconds, and consumed 1464.80 MB of memory. Double the size of the dataset to 50%, the balanced dataset trained in 210.02 seconds, tested in 3.80 seconds, and consumed 1673.21 MB of memory. The unbalanced dataset of the same size, however, trained in 26.14 seconds, tested in 5.78 seconds, and consumed 1873.63 MB of memory.

At 75% dataset, the balanced dataset used 191.26 seconds to train, 5.05 seconds to test, and 2012.68 MB of memory. Its unbalanced counterpart, however, used a total of 173.94 seconds and 1873.63 MB of memory. Then, at full 100% sample size, the balanced dataset used the most, with its training taking 217.96 seconds in total and 2042.36 MB of memory. The unbalanced dataset at the same size, however, took 194.16 seconds to complete its process while using 1869.68 MB of memory, which shows lower computational requirements. These findings, as Figure 4 depicts, clearly indicate that although larger and well-balanced datasets correspond to higher processing time and memory consumption, they are typically found to provide better model performance compared to unbalanced datasets.

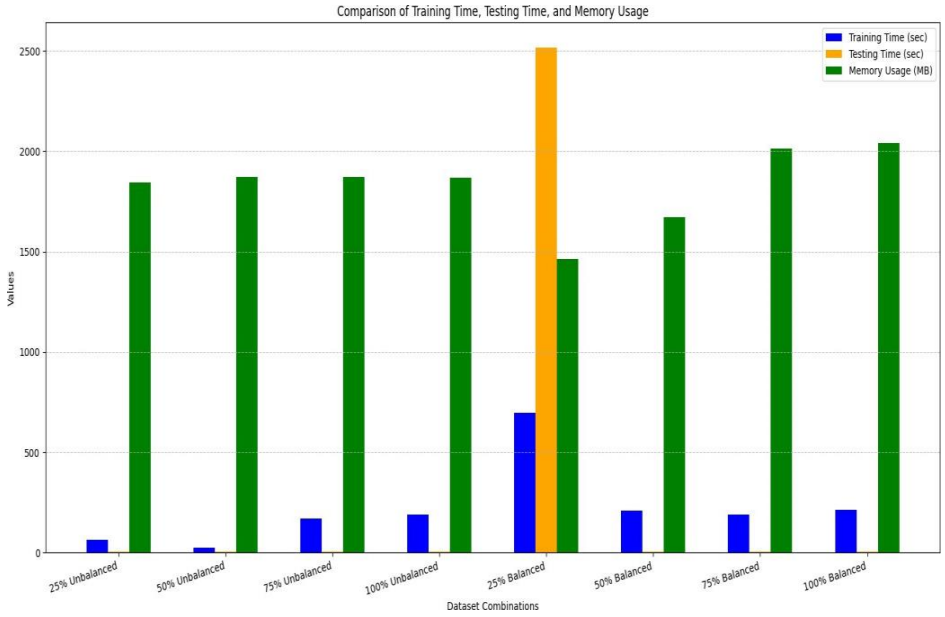

Figure 4.Resource Usage Comparison for Balanced and Unbalanced Datasets at Various Dataset Proportions

Figure 4 illustrates the trend is consistent across other dataset proportions, as reflected in Table 4. Additionally, at each division balanced datasets consume more resources and memory compared to the unbalanced dataset. An unusual spike in the testing time was observed in the 25% balanced dataset case. Through this, the overhead that is being faced to process the same number of samples for every class using augmentation techniques caused greater fluctuation at test time and led to longer evaluation time. In contrast, the 25% unbalanced dataset had a skewed distribution dominated by majority class samples, resulting in less computational complexity and faster testing. At higher dataset sizes such as 75% and 100%, this testing time anomaly did not occur because GPU memory was utilized more efficiently, and batch processing became more stable due to the larger number of samples, even in unbalanced conditions. This improved system-level optimization helped stabilize testing time at larger scales regardless of class balance.

Further experiments were conducted using various training and testing split ratios, including 50%-50%, 60%-40%, 70%-30%, 80%-20%, and 90%-10%, as detailed in Table 6. The corresponding images number allocated to each split configuration is listed in Table 5. The results from these tests indicate that the model achieved its best performance when trained with 70% of the data and tested on the remaining 30%. Specifically, with 42,000 images used for training and 18,000 for testing (as outlined in Table 5), the model attained its highest accuracy of 80.93%, along with a precision of 0.8098, recall of 0.8093, and a ROC-AUC score of 0.89.This superior performance at the 70%-30% split is attributed to the model receiving an optimal amount of training data, allowing it to effectively learn meaningful patterns while still having a sufficiently large and diverse testing set to reliably evaluate its performance. The dispersion of images across different train-test split ratios is provided in Table 5 for reference.

Table 5. Number Images Across Different Training-Testing Ratios

Split Ratio	Training Images	Test Images	Total Images
50%-50%	30,000	30,000	60,000
60%-40%	36,000	24,000	60,000
70%-30%	42,000	18,000	60,000
80%-20%	48,000	12,000	60,000
90%-10%	54,000	6,000	60,000

The corresponding performance metrics are reported in Table 6**.**

Table 6.Performance Metrices Across Different Training-Testing Ratios

Ratio	Accuracy	Precision	Recall	Support	ROC
50% Training 50% Testing	78.56%	0.7901	0.7856	5000.0	0.88
60% Training 40% Testing	78.48%	0.7876	0.7848	10000.0	0.88
70% Training 30% Testing	80.93%	0.8098	0.8093	10000.0	0.89
80% Training 20% Testing	80.19%	0.8071	0.8019	10000.0	0.79
90% Training 10% Testing	79.76%	0.8009	0.7976	10000.0	0.89

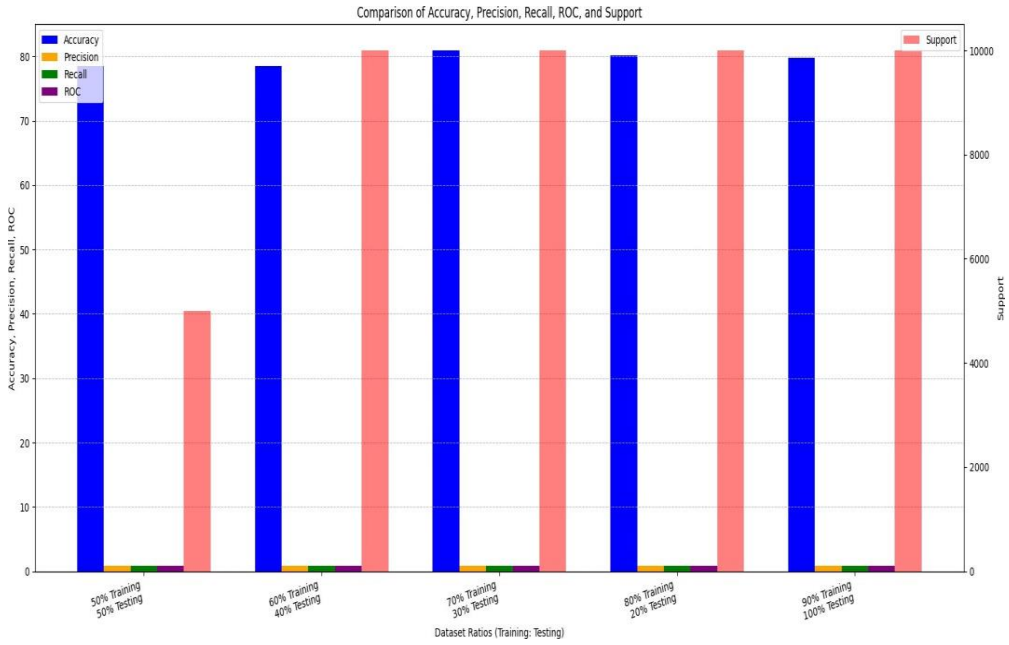

Figure 5.Impact of Training-Testing Splits on Performance Metrics of GoogleNet Model

Figure 5 shows the performance of the model using various training and testing split ratios. When it was tested using the highest accuracy, the 70%-30% train-test split was used. Using this ratio, the model achieved a reasonable amount of resource consumption, which included 243.72 seconds and 3.59 seconds of training and testing time, respectively, and 3171.60 MB of memory consumption, as shown in Figure 6. This is because this ratio strikes an optimal balance between having sufficient data to enable the model to learn sufficiently and sufficient unseen data to adequately test its generalization ability.

The results also show that when the proportion of training data is too great compared to test data, the model overfits too specific in the training set and performing poorly on new, unseen examples. Overfitting was seen at the 90%-10% split, at which the model's performance dramatically dropped. At lower proportions like 50%-50% and 60%-40%, the model underperforms its potential due to a lack of training data. Notably, performance at the 80%-20% split was actually below optimal compared to these lower proportions, suggesting that an equal division between training and test data is required for optimal performance.

Table 7.Resource Usage Metrices across Different Training-Testing Ratios

Ratio	Training Time	Testing Time(s)	Memory(MB)
50% Train, 50% Test	267.53	1.81	3122.02
60% Train, 50% Test	216.99	4.25	3155.76
70% Train, 50% Test	243.72	3.59	3171.60
80% Train, 50% Test	251.05	3.83	1695.29
90% Train, 50% Test	326.41	3.76	1709.27

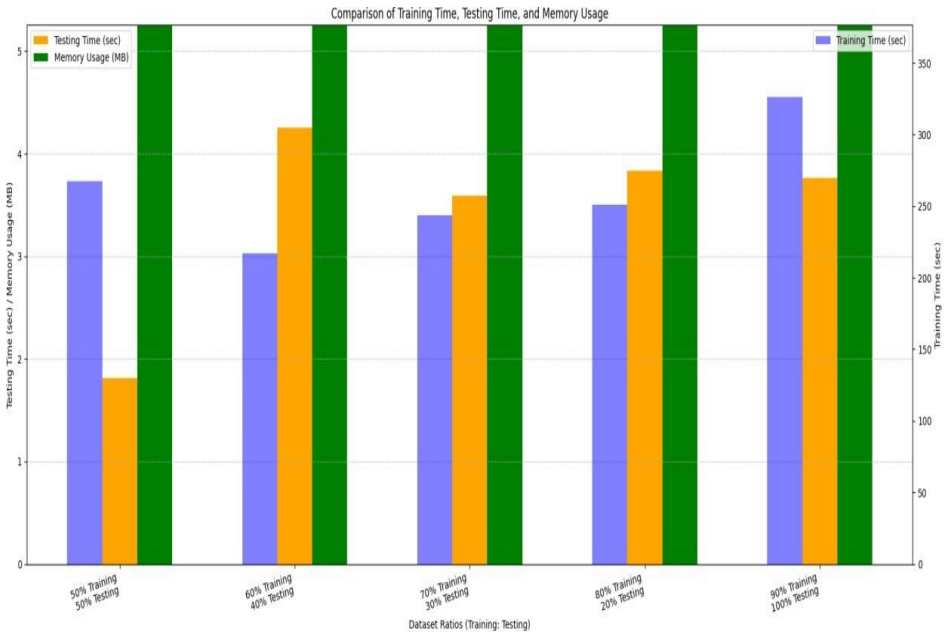

Figure 6.Training Time, Testing Time, and Memory Usage Across Different Train-Test Split Ratios

6. RESULT DISSCUSSION

The results presented in Table 3 and visualized in Figures 2 and 3 demonstrate a clear trend: balanced datasets consistently achieve higher accuracy and better evaluation metrics than unbalanced datasets across all dataset proportions**.** This consistent superiority can be attributed to the equal distribution of class samples, which mitigates the risk of majority class bias and allows the model to learn representative patterns for all classes. In contrast, in unbalanced datasets, the number of samples per class is unequal, causing the model to become overly specialized towards the majority class. This leads to poor generalization on unseen data. Therefore, the experimental results show superior performance on each balanced dataset compared to unbalanced datasets, as shown in Figure 2-3.

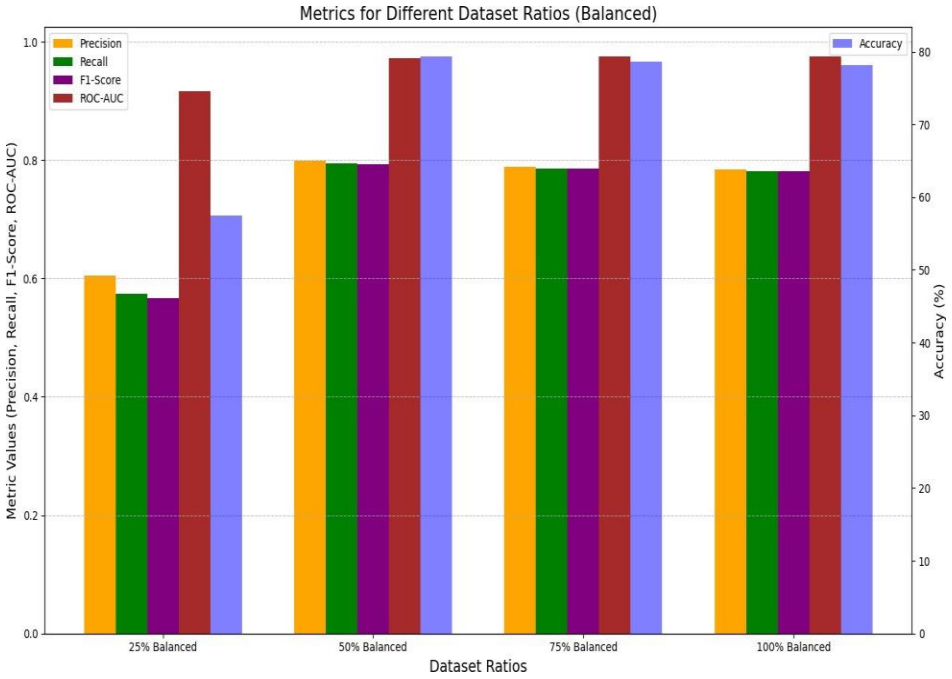

Figure 7.Evaluation Metrics Comparison for Balanced CIFAR-10 Datasets at Different Proportions

The bar plots in Figure 7 show a comparison of the evaluation metrics precision, recall, F1-score, ROC-AUC, and accuracy for different sizes of balanced datasets. Similarly, Figure 8 shows the metrics for corresponding unbalanced datasets. For easy visulization of the comparatively higher accuracy values than the other metrics, a secondary y-axis is used for better clarity. The plots show that larger unbalanced datasets, particularly at the 75% and 100% proportions, tend to yield better overall performance. At these sizes, accuracy achieves the highest of 77.91%, while precision, recall, and F1-score values reach 0.78.

Moreover, it is clear model performance is enhanced as the dataset size is enlarged, as seen from the results listed in Table 3. This is due to the fact that machine learning as well as deep learning models will tend to perform optimally when they are fed more data, as this enables them to learn patterns and generalize efficiently. The optimal performance of the model is achieved by using the 100% dataset, which gives a greater number of examples than the 75%, 50%, and 25% datasets clearly visible in Figure 7.

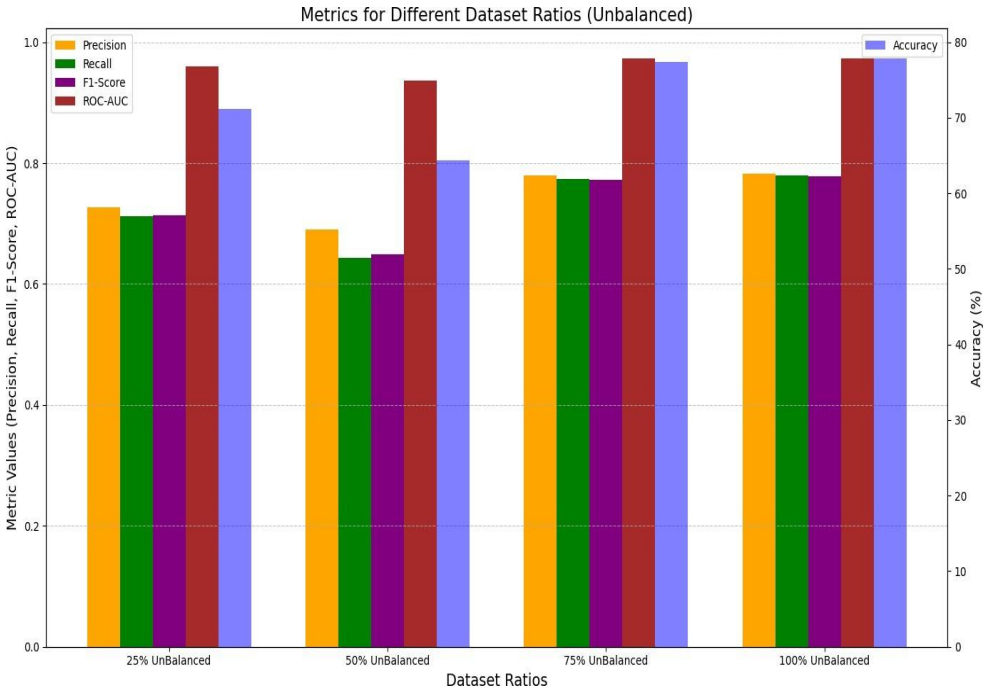

Figure 8.Evaluation Metrics Comparison for Unbalanced CIFAR-10 Datasets at Different Proportions

The other important finding of this study is that the size of the dataset is greater, and thus the training and testing time and memory usage increase. This result is expected because analyzing a greater amount of data necessarily requires more processing resources, i.e., time and memory, to process, train, and test models. The model will require more time and memory to perform these operations with more examples. Hence, the whole 100% dataset takes the greatest processing time and memory among the smaller 25%, 50%, and 75% datasets, as illustrated in Figure 8.

Furthermore, testing with varied training and testing split ratios (50%-50%, 60%-40%, 70%-30%, 80%-20%, and 90%-10%) also demonstrated that the model performed optimally at a 70%-30% split, as indicated in Table 6 and Figure 5. This is because this split offers an even split with enough data for the model to learn adequately while having enough testing data to accurately measure performance without overfitting. With increased training ratios like 80%-20% and especially 90%-10%, there is a possibility of overfitting since the model becomes too specific to the training data, reducing its ability to generalize to new, unseen data. Lower ratios like 50%-50% and 60%-40%, on the other hand, leave the model with less data to learn from, reducing its ability to learn and hence performing worse compared to the optimal 70%-30% split.

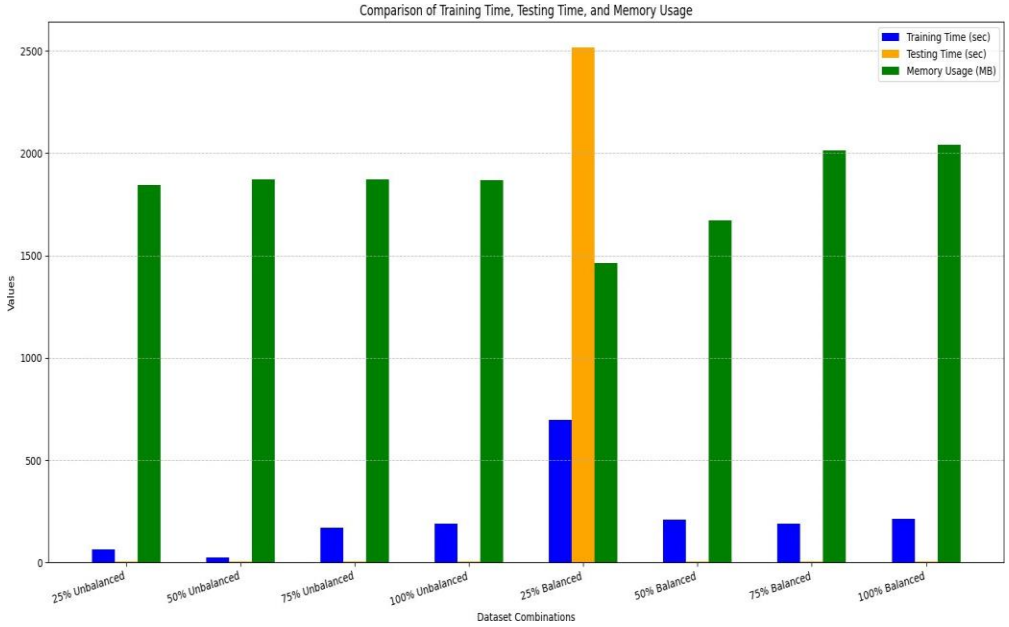

Figure 9.Training and Testing Time Analysis Across Dataset Proportions for Balanced and Unbalanced Data

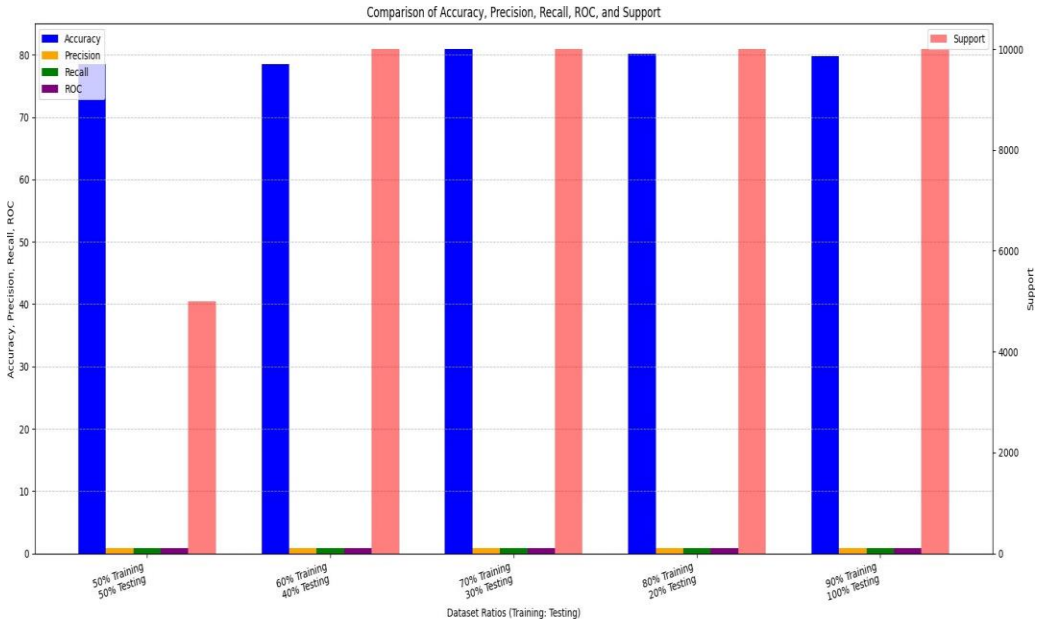

Figure 10.Relationship Between Training/Testing Time and Performance Metrics Across Different Dataset Proportions

A comparison of training and test times for balanced and unbalanced datasets with varying dataset sizes is presented in Figure 9. Figure 10 also presents the correspondence between training/testing times and performance measures against different dataset ratios. The trends are seen to hold true, as indicated by Table 4 and Table 6. In the plots, important evaluation metrics like Accuracy, Precision, Recall, and ROC-AUC are marked on the main y-axis, whereas the values for support are marked along a secondary y-axis. Dual y-axes utilize better clarity by scaling both smaller and larger values suitably so that the main metrics are easily readable while presenting the support values visibly along with them.

It should be noted that the performance trends of GoogleNet in this study are linked to the new features of its architecture. In contrast to other standard CNN architectures, GoogleNet contains Inception modules which enable the network to extract multi-scale features using parallel convolutions with various kernel sizes. This architecture enhances the capacity of the model to learn various patterns in an image but also renders the model class-imbalanced. In situations where over-sampling of some classes occurs, the multi-scale feature extractors will favor dominant class patterns at the expense of the model's capacity to generalize. But with the growth in the size of the overall dataset, minority class instances also grow, subjecting the Inception modules to a higher variety of patterns to learn. The non-proportional growth in data variety acts to counterbalance the learning process, reducing the majority class bias and improving generalization. Observe that this effect can be reversed in other architectures, like ResNet, which uses residual connections in deep sequential layers, or less complex models like VGG, which rely on simple sequential feature extraction routes. These architectural differences dictate the way models respond to problems like dataset size and class imbalance, and this leads to the performance trends observed in this work.

7. Conclusion

This research involved a series of experiments with GoogleNet pre-trained model to assess its performance on the CIFAR-10 data under varying conditions. The four dataset ratios (25%, 50%, 75%, and 100%) were tested under unbalanced and balanced datasets. The results always indicated that the model performed optimally when it was trained under balanced datasets regardless of the dataset size. Furthermore, the growth of the dataset size resulted in stepwise improvement in model performance, with optimum accuracy on the whole 100% balanced dataset. The study also explored the effect of different training and test data ratios, including 50%-50%, 60%-40%, 70%-30%, 80%- 20%, and 90%-10%. Among these, the 70%-30% ratio produced the most optimal results, offering a good balance of adequate training data and adequate test data to identify generalization. Generally, the results show that good model performance on image classification problems with GoogleNet is a function of a combination of a few important factors: use of a right balance and adequately sized dataset, proper hyperparameter tuning, and specification of right ratio for train-test split.

8. Acknowledgements

The authors sincerely appreciate the support and assistance received during the course of this research. Gratitude is extended to the academic supervisors and technical staff for their helpful guidance and encouragement. The authors would also like to acknowledge the availability of open-source tools and resources, which greatly contributed to the successful execution of this study.

References

[1] S. Serte, A. Serener, and F. Al-Turjman, "Deep learning in medical imaging: A brief review," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 10, pp. e4080, 2022.

[2] J. U. M. Akbar, S. F. Kamarulzaman, A. J. M. Muzahid, M. A. Rahman, and M. Uddin, "A comprehensive review on deep learning assisted computer vision techniques for smart greenhouse agriculture," *IEEE Access*, vol. 12, pp. 4485-4522, 2024.

[3] C. Rui, "Research on classification of cross-border e-commerce products based on image recognition and deep learning," *IEEE Access*, vol. 9, pp. 108083-108090, 2020.

[4] X. Zhang, N. Han, and J. Zhang, "Comparative analysis of VGG, ResNet, and GoogLeNet architectures evaluating performance, compu- tational efficiency, and convergence rates," *Appl. Comput. Eng.*, vol. 44, pp. 172-181, 2024.

[5] M. Wojciuk, Z. Swiderska-Chadaj, K. Siwek, and A. Gertych, "Im- proving classification accuracy of fine-tuned CNN models: Impact of hyperparameter optimization," *Heliyon*, vol. 10, no. 5, 2024.

[6] A. Ali, S. M. Shamsuddin, and A. L. Ralescu, "Classification with class imbalance problem," *Int. J. Adv. Soft Comput. Appl.*, vol. 5, no. 3, pp. 176-204, 2013.

[7] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321-357, 2002.

[8] D. Soekhoe, P. Van Der Putten, and A. Plaat, "On the impact of dataset size in transfer learning using deep neural networks," in *Proc. Int. Symp. Intell. Data Anal.*, Cham: Springer, 2016, pp. 50-60.

[9] H. Bichri, A. Chergui, and M. Hain, "Investigating the impact of train/test split ratio on the performance of pre-trained models with custom datasets," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 2, 2024.

[10] A. Gosain and S. Sardana, "Handling class imbalance problem using oversampling techniques: A review," in *Proc. Int. Conf. Adv. Comput., Commun. Inform. (ICACCI)*, 2017, pp. 79-85.

[11] A. Zawadzki, M. Karpin´ski, and M. Piechocki, "Logo and brand recog- nition from imbalanced dataset using MiniGoogLeNet and MiniVGGNet models," in *Proc. 11th Asian Conf. Intell. Inf. Database Syst. (ACIIDS)*, Springer, 2019, pp. 374-383.

[12] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015, pp. 1-9.

[13] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 843-852.

[14] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Netw.*, vol. 106, pp. 249-259, 2018.

[15] V. S. Spelmen and R. Porkodi, "A review on handling imbalanced data," in *Proc. Int. Conf. Curr. Trends Converg. Technol. (ICCTCT)*, 2018, pp. 1-11.

[16] M. S. Shelke, P. R. Deshmukh, and V. K. Shandilya, "A review on imbalanced data handling using undersampling and oversampling technique," *Int. J. Recent Trends Eng. Res.*, vol. 3, no. 4, pp. 444-449, 2017.

[17] I. Muraina, "Ideal dataset splitting ratios in machine learning algorithms: General concerns for data scientists and data analysts," in *Proc. 7th Int. Mardin Artuklu Sci. Res. Conf.*, 2022, pp. 496-504.

[18] Unnisa, Z., Tariq, A., Sarwar, N., Din, I., Serhani, M. A., & Trabelsi, Z. (2025). Impact of fine-tuning parameters of convolutional neural network for skin cancer detection. Scientific Reports, 15(1), 1-23.

[19] Shah, S., Sarwar, N., Salam, A., Amin, F., Ullah, F., Khan, A., ... & Garay, H. (2025). A flexible and lightweight signcryption scheme for underwater wireless sensor networks. Scientific Reports, 15(1), 13511.

[20] Sarwar, N., Al-Otaibi, S., & Irshad, A. (2025). Optimizing Breast Cancer Detection: Integrating Few-Shot and Transfer Learning for Enhanced Accuracy and Efficiency. International Journal of Imaging Systems and Technology, 35(1), e70033.

[21] Ayub, N., Sarwar, N., Ali, A., Khan, H., Din, I., Alqahtani, A. M., ... & Ali, A. (2025). Forecasting Multi-Level Deep Learning Autoencoder Architecture (MDLAA) for Parametric Prediction based on Convolutional Neural Networks. Engineering, Technology & Applied Science Research, 15(2), 21279-21283.

[22] Sarwar, N., Irshad, A., Naith, Q. H., D. Alsufiani, K., & Almalki, F. A. (2024). Skin lesion segmentation using

deep learning algorithm with ant colony optimization. BMC Medical Informatics and Decision Making, 24(1), 265.

[23] Wang, Y., Rajkumar Dhamodharan, U. S., Sarwar, N., Almalki, F. A., & Naith, Q. H. (2024). A hybrid approach for rice crop disease detection in agricultural IoT system. Discover Sustainability, 5(1), 99.

[24] Ullah, R., Yahya, M., Mostarda, L., Alshammari, A., Alutaibi, A. I., Sarwar, N., ... & Ullah, S. (2024). Intelligent decision making for energy efficient fog nodes selection and smart switching in the IOT: a machine learning approach. PeerJ Computer Science, 10, e1833.

[25] Akram, A., Rashid, J., Hajjej, F., Yaqoob, S., Hamid, M., Arshad, A., & Sarwar, N. (2023). Recognizing Breast Cancer Using Edge-Weighted Texture Features of Histopathology Images. Computers, Materials & Continua, 77(1).

[26] "The CIFAR-10 dataset," Available: https://www.kaggle.com/c/cifar-10? utm_source