JOURNAL OF COMPUTING HORIZONS
Vol. 01, No. 01, 2026
DOI: 10.22610/JCH.V1I1.4991

RESEARCH ARTICLE | PEER REVIEWED

A Systematic Investigation of Xception for Melanoma Classification Across Data Imbalance and Training Configurations

Naila Batool [1] *, Aqsa Akram [1], Amna Zaheer [1]
[1] *Department of Computer Science, University of Management and Technology, Lahore (Sialkot Campus), Pakistan*
* **Corresponding Author:** nailabatool617@gmail.com

ABSTRACT – Deep learning has demonstrated considerable potential for automated skin lesion classification; however, its performance is influenced by dataset characteristics, class distribution, and model configurations. This study presents an experimental evaluation of the Xception architecture for melanoma skin cancer classification, investigating the effects of dataset balancing, data proportions, training–testing ratios, and hyperparameter configurations on classification performance and computational efficiency. Balanced and imbalanced datasets are evaluated at data proportions of 25%, 50%, 75%, and 100%, while five training–testing ratios ranging from 50:50 to 90:10 are investigated. We also examine different hyperparameter configurations to assess their influence on model performance. Classification effectiveness is evaluated using accuracy, precision, Recall, F1-score, and ROC-AUC, while training time, testing time, memory consumption, and total execution time are considered to assess computational requirements. Experimental results indicate that balanced datasets generally provide more consistent classification performance across the evaluated metrics. Among the investigated training–testing ratios, the 80:20 configuration achieves the highest accuracy of 86.31%, F1-score of 85.35%, and ROC-AUC of 93.88%. Furthermore, the results highlight the relationship between training data availability, classification performance, and computational cost. This study provides a comprehensive experimental assessment of Xception and highlights the importance of dataset composition and experimental configuration in melanoma classification.

Keywords: *Melanoma classification; Xception; deep learning; class imbalance; dataset proportions; training–testing ratio; computational efficiency.*

I. INTRODUCTION

Melanoma is a particularly aggressive variant of cutaneous cancers requiring early detection to improve patient prognosis [1, 2]. Dermoscopy provides valuable information about the visual characteristics of cutaneous lesions during clinical evaluations. Diagnosing melanoma, however, is challenging due to significant intra-class differences in lesion appearance, pigmentation patterns, morphological characteristics, and texture [3, 4]. Several recent publications explore the feasibility of artificial intelligence-based automatic skin lesion classification using deep learning algorithms [5, 6].

Deep Convolutional Neural Networks show substantial promise for automatically extracting features from medical images in clinical applications [7, 8]. One example of a CNN architecture is the Xception network, which uses depthwise separable convolutions to automatically extract features while minimizing computational cost compared to traditional convolution layers [9, 10]. While deep learning models have been successfully used to classify skin lesions, their accuracy depends significantly on various experimental parameters, such as the data composition, the number and balance of the different classes, the size of the training sets, and the architectural choices made when designing deep learning networks [11, 12, 13].

Another problem to be considered is the presence of class imbalance in melanoma diagnoses because unequal numbers of samples will impair model performance for minority classes [14, 15]. Additionally, modifying the proportions of training and testing data may affect classification performance [16, 17, 18]. Moreover, varying the split ratio between training and testing datasets influences classification results [16, 17, 18]. An increase in training data does not always improve classification accuracy; it depends on the evaluation metrics used [19, 20]. The computational cost, including training, inference, and memory overhead, is an essential factor in evaluating the practical applicability of deep neural networks [21]. As such, evaluating classification performance alongside computational cost across different experimental settings is necessary to gain a holistic view of the model.

Driven by these observations, we conduct experiments investigating the effect of balanced and imbalanced data splits at various proportions, explore the effect of split ratios between training and testing sets of 50:50, 60:40, 70:30, 80:20, and 90:10, investigate the training process costs, and examine the effects of hyperparameter tuning on classification performance. This investigation evaluates classification performance using accuracy, precision, Recall, F1-score, and ROC-AUC, while computational costs include training time, inference time, and memory utilization. Some main findings of this investigation include the following points:

- Xception is evaluated on balanced and imbalanced datasets at data proportions of 25%, 50%, 75%, and 100% to investigate the influence of class distribution and dataset availability on classification performance.
- Five different training–testing configurations (50:50, 60:40, 70:30, 80:20, and 90:10) are investigated. The 80:20 ratio yields the highest accuracy, F1-score, and ROC-AUC (86.31%, 85.35%, and 93.88%, respectively) among the ratios tested.
- Training time, testing time, memory consumption, and total execution time are analyzed to examine the computational requirements of Xception under different experimental configurations.
- Multiple classification metrics are considered to assess predictive performance and identify variations attributable to dataset distribution and training data availability.

This paper is structured into sections as outlined below. Section II introduces the data used and the methods employed in the experiments. Section III analyzes the methodology, Section IV discusses the experimental setup, whereas Section V discusses those outcomes. The conclusion is provided in Section VI.

II. RELATED WORK

Deep learning techniques have been widely explored for automated skin lesion classification, particularly convolutional neural networks (CNNs), transfer learning approaches, and hybrid models. Xception employs depthwise separable convolutions to extract features from images with minimal computational cost. In research using the Xception network in combination with various pre-trained CNN models on the HAM10000 dataset, Mahmud et al. [1] achieved an accuracy of 88.72% and a weighted F1 score of 0.89. Furthermore, Shah et al. [10] incorporated the Xception model with particle swarm optimization and subspace KNN classifiers, achieving 98.5% accuracy on the ISIC 2018 dataset. Other studies considered different architectures, including CNN-Vision Transformer fusion [11], Vision Transformer-based classification [3], and SynthraXCoreNet ensemble [8]. Other investigations examined different methodologies, highlighting developments within deep learning approaches for skin lesion analysis.

Class imbalance is another important issue, given that most skin lesion datasets exhibit a high discrepancy in the number of samples per category. Many studies have resolved this problem through class balancing, resampling, and augmentation techniques. Class weighting and stratified sampling were used by Cherukuri and Vara Prasad [5]; federated learning and explainable AI approaches, and strategies that address class imbalance issues were tested by Jaya et al. [14], whereas adversarial domain adaptation was explored by Gilani et al. [13]. Bias analysis in skin lesion classifiers was conducted by Corbin et al. [16]. Other studies investigated model reliability and generalizability, particularly across different distribution datasets, such as metric-based few-shot learning [12], domain-adaptive Vision Transformer ensembles [17], and active domain adaptation [18]. These studies demonstrate the significance of the dataset composition and distribution when evaluating the performance of skin lesion classifiers.

Model configuration and optimization also influence classification performance. Raza et al. [2] investigated optimization strategies and reported strong results using Adam with a customized activation function. Mahmud et al. [1] and Malhotra and Kaur [9] also examined pretrained CNNs and optimization configurations for skin lesion classification. Priambodo and Fatichah [15] investigated the application of lightweight CNN models enhanced by the Convolutional Block Attention Module (CBAM). Benaly et al. [7] analyzed Grad-CAM and Grad-CAM++, while Jadhav et al. [19] used Grad-CAM, SHAP, and LIME to explain their models. Both studies have made complementary contributions regarding optimization, feature representation, and interpretability within artificial intelligence applications.

Computational efficiency is another critical factor, especially when computing resources are constrained. Vinitha et al. [21] studied the application of ensemble lightweight CNN models in edge computing, while Lubis et al. [20] applied ReXNet-150 to accelerate and improve melanoma detection efficiency. Kollem et al. [4] developed a novel framework called SKIN-MARS, which uses Multi-Scale Swin Transformer networks for skin lesion segmentation, demonstrating the high computational expense of such analysis tasks. Despite recent progress, many studies focus on specific architectures, dataset configurations, and performance metrics. Systematic evaluation of Xception across varying dataset proportions, different class distributions, various training-testing splits, and computational metrics has not yet been performed. Therefore, this work will experimentally evaluate Xception for melanoma skin cancer classification by examining classification performance across several datasets and estimating training time, testing time, memory usage, and execution time.

III. METHODOLOGY

This section presents the methodology using the Xception pretrained model. The methodology begins with details of the datasets used in the experiments, followed by tables summarizing general and experiment-specific hyperparameters. A methodology diagram, shown in Figure 1, visually illustrates the process flow throughout model training and evaluation.

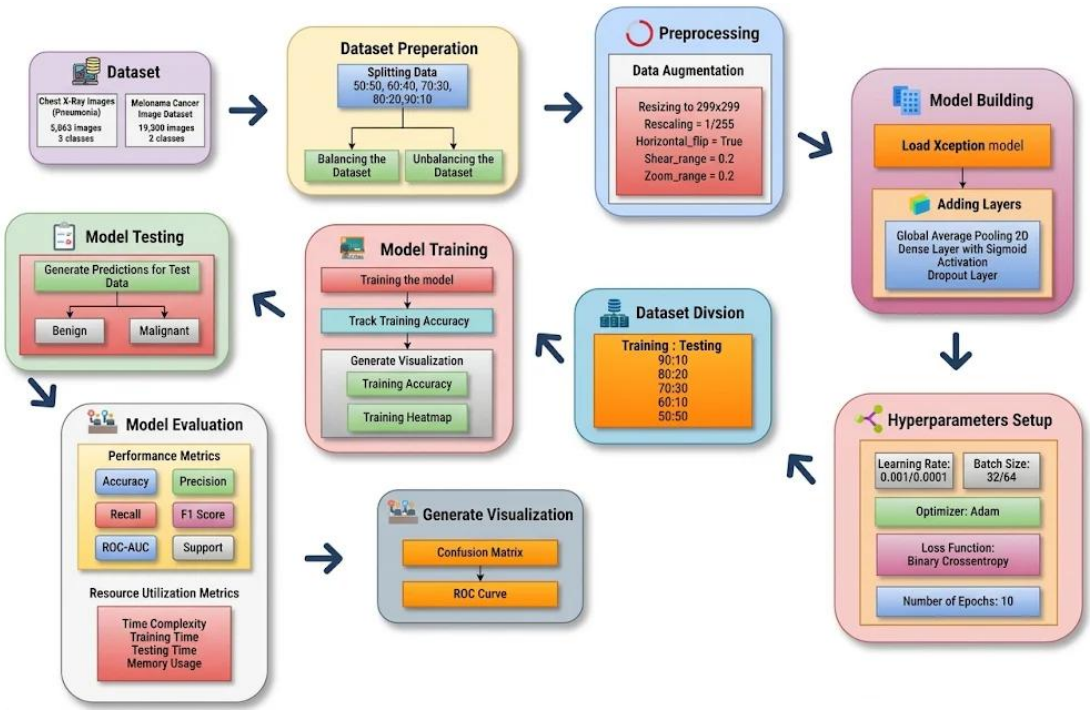


FIGURE 1: Methodology Diagram

Figure 1 outlines the overall process used in the experiment; the methodology starts with the selection of datasets, with two datasets loaded into memory to make them accessible for model training. After loading, the datasets were split into four combinations of different data sizes: 25%, 50%, 75%, and 100%. These combinations were then processed under two scenarios: unbalanced and balanced datasets. For each of these scenarios, the datasets were further divided into different training and testing ratios, including 20:80, 50:50, 70:30, and 80:20. By performing this step, the developed model was tested on different sizes of datasets to assess its robustness and adaptability towards different class distributions and splits for training/testing. After training, the model was tested on the test set, which classified the pictures as either benign or malignant. Evaluation metrics such as accuracy, precision, Recall, F1 score, ROC-AUC, and support were computed to assess the model's performance. The evaluation of performance involves tracking time complexity, training time, and memory usage to provide a complete picture of the model's efficiency and effectiveness.

A. Datasets Description

Melanoma Skin Cancer Dataset is an organized collection of 13900 RGB images sourced from Kaggle, uniformly resized to 224x224 pixels in JPEG format, totaling 83.27 MB. The dataset was created to facilitate improvements in diagnosis by leveraging advanced machine-learning models in dermatology. The images in the dataset are distributed into two categories: Benign and Malignant. These two classes are further divided into two subsets: training and testing. The training set contained 6289 images in the benign class and 5590 in the malignant class. Likewise, the testing set consists of 1000 benign images and 1000 malignant images. The high-quality images make it a useful resource for early detection and accurate diagnosis in melanoma classification, as shown in Figure 2.

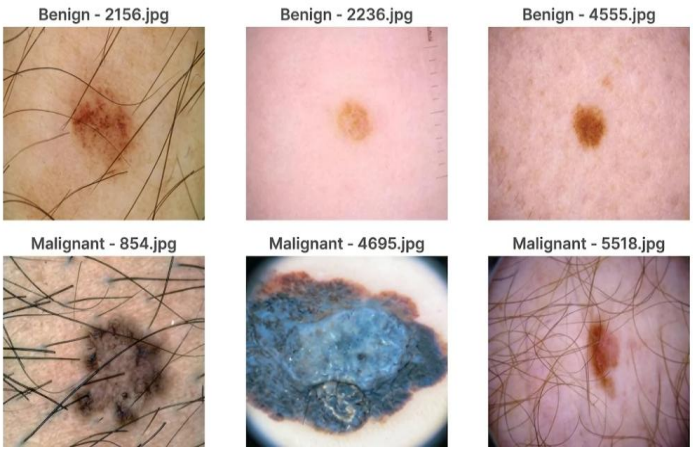


FIGURE 2: Melanoma Cancer Image Dataset

B. Experimentation in Balanced and Unbalanced Datasets

For this experiment, two combinations of the datasets were made as balanced and unbalanced. The following Table 1 presents the distribution of images in each class (Benign and Malignant) for both balanced and unbalanced datasets across various dataset sizes (25%, 50%, 75%, and 100%).

TABLE 1: Distribution of Balanced and Unbalanced Datasets

Dataset Size	Train Subset (Balanced)	Test Subset (Balanced)	Train Subset (Unbalanced)	Test Subset (Unbalanced)
25% Dataset	Benign: 1572, Malignant: 1397	Benign: 250, Malignant: 250	Benign: 628, Malignant: 838	Benign: 150, Malignant: 100
50% Dataset	Benign: 2795, Malignant: 2795	Benign: 500, Malignant: 500	Benign: 1886, Malignant: 1118	Benign: 200, Malignant: 300
75% Dataset	Benign: 4192, Malignant: 4192	Benign: 750, Malignant: 750	Benign: 1886, Malignant: 2515	Benign: 450, Malignant: 300
100% Dataset	Benign: 5590, Malignant: 5590	Benign: 1000, Malignant: 1000	Benign: 3773, Malignant: 2236	Benign: 400, Malignant: 600

Figure 3 shows a comparison between the train and test datasets. The distribution of Benign and Malignant images across balanced and unbalanced datasets for different sizes (25%, 50%, 75%, and 100%).

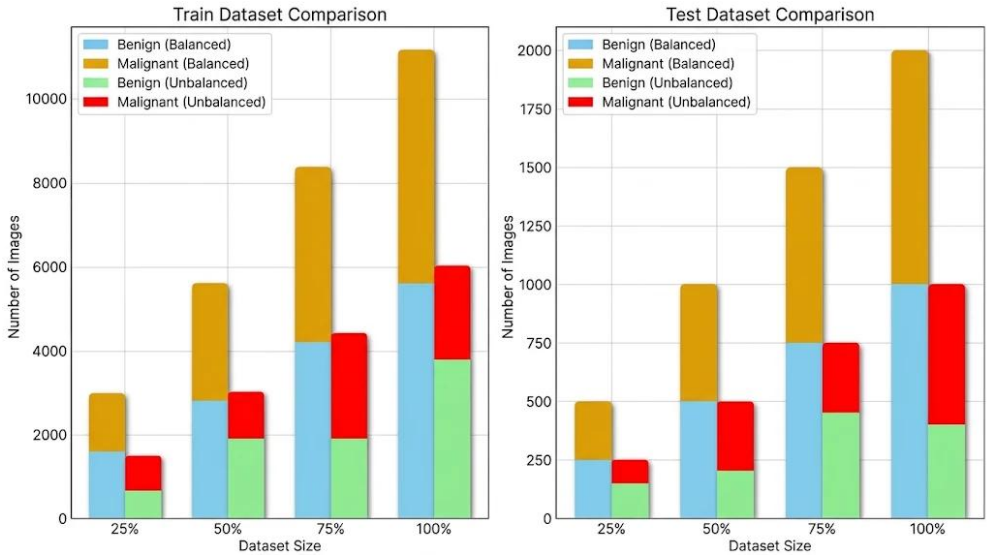


FIGURE 3: Train and Test Dataset Comparison

C. Method to Balance the Dataset

In balancing the dataset, first, to assess the class imbalance, the number of images in each class was counted. After counting, images were randomly selected and copied from the original dataset into a new directory to create a subset of the dataset for training and testing. Next, to balance the dataset, a minimum class size was determined as a reference. Using this reference, random images were selected from each class to match the smaller class's size. Lastly, the images of both classes were balanced to ensure a balanced dataset. The following steps outline the process of splitting and also balancing the dataset. A Process of splitting the dataset (in this case, 50%) is shown in Figure 4 and Figure 5.

D. Method to Unbalance the Dataset

To create an unbalanced dataset, the first step is to create a subset by randomly selecting images from each class. Different fractions of images were selected for each class to induce the class imbalance. According to the selected fractions, images were distributed to each class to create an imbalanced dataset. For storing these imbalanced subsets, separate directories were created. Lastly, the images of both classes were stored to ensure class imbalance. The following steps outline the process of splitting and balancing the dataset. A Process of splitting the dataset: 50% dataset, setting fractions as 60:40 for training and 40: 60 for testing classes (Benign: Malignant) shown in Figure 6.

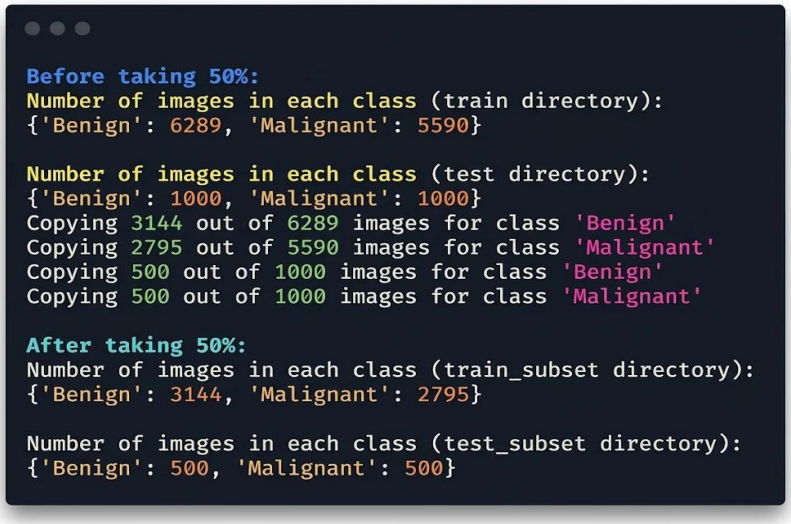


FIGURE 4: Splitting Dataset into 50%



FIGURE 5: 50% Balanced Dataset

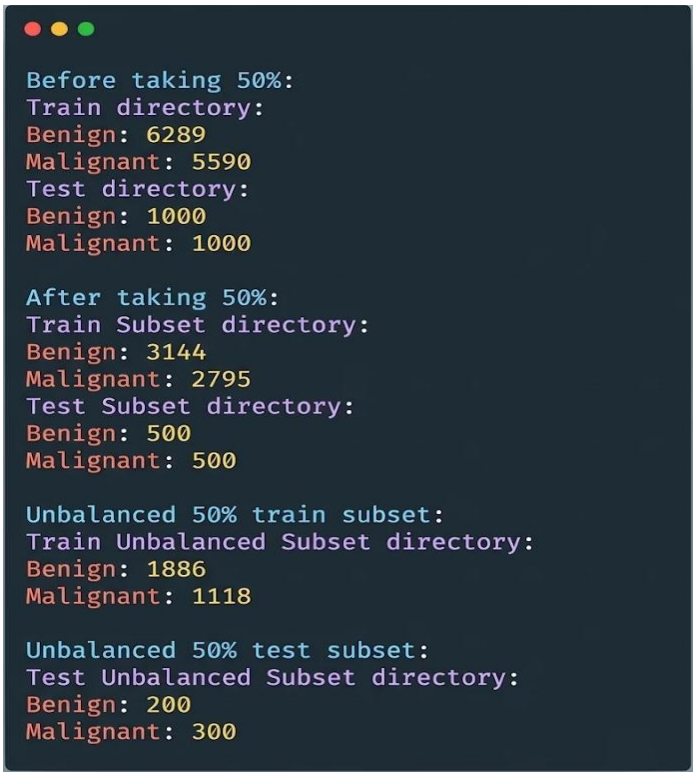


FIGURE 6: Process of Unbalancing the Dataset

IV. EXPERIMENTAL SETUP

A. Configuration and Tools

The computational analysis used a hybrid approach comprising local and cloud computing. For the local machine, the simulation was run using Microsoft Windows 10 Pro (x64) with an Intel Core i5-6200U processor and 8 GB of RAM. In contrast, the cloud computing part used Google Colab for free, which offers a T4 GPU, 97.0 GB of disk space, and 12 GB of RAM – adequate for deep learning computations. Python 3.8 was chosen for its compatibility with deep learning frameworks. Two datasets utilized during this study were sourced from Kaggle repositories [1, 2].

B. Experimentation Hyperparameters

The hyperparameters fine-tuned for the experiments, along with their values, are listed in Table 2. From this table, two hyperparameters were selected for tuning: Learning Rate (0.001, 0.0001) and batch Size (32, 64). The combination LR = 0.001 and batch = 32 strikes an optimal compromise between efficiency and stability during training. Combination 2 (LR=0.0001, Batch=64) optimizes the model's ability to accelerate training. Table 3 shows the results of two different hyperparameter combinations utilizing different metrics.

TABLE 2: Experimentation Hyperparameters

Hyperparameter	Value
Learning Rate	0.001, 0.0001
Batch Size	32, 64
Epochs	10
Optimizer	Adam
Loss Function	Binary Crossentropy
Data Augmentation	Resizing: 299x299, Zoom Range: 0.2, Rescaling: 1/255, Shear Range: 0.2, Horizontal Flip: True
Base Model	Xception (Pretrained on ImageNet)

Specifically, with a learning rate of 0.001 and a batch size of 32, the neural network achieves an accuracy of 87.93%. On the other hand, the second combination, with LR = 0.0001 and batch size = 64, achieves an accuracy of 84.02%. Resource requirements differ substantially; thus, combination 1 spends 3031.73 seconds for training and 20.60 seconds for testing, whereas combination 2 takes 3043.21 seconds to complete the training stage & 15.46 seconds to test its performance. Memory consumption for combination 1 amounts to 2108.95 megabytes, while for combination 2 – 2406.97 megabytes. As far as accuracy scores go, however, combination 2 offers better stability during training and improved generalization at the expense of extended training duration.

Furthermore, combination 1 converges much faster and achieves higher training accuracy but exhibits severe overfitting and insufficient generalization to the validation dataset. Combination 2 shows superior stability, better generalization, and a lower tendency towards overfitting, although its training takes significantly longer to complete. Overall, combination 2 strikes a balance between stability and generalization, yielding higher accuracy on novel datasets.

TABLE 3: Different Settings of Hyperparameters

Metrics Combination	Combination 1 (LR=0.001, Batch=32)	Combination 2 (LR=0.0001, Batch=64)
Accuracy	87.93%	84.02%
Training Time	3031.73 seconds	3043.21 seconds
Testing Time	20.60 seconds	15.46 seconds
Memory Usage	2108.95 MB	2406.97 MB
Time Complexity	$O(1.29 \times 1012)$ operations	$O(1.29 \times 1012)$ operations

V. RESULTS AND DISCUSSION

A. Results Based on Performance Metrics

The results in Table 4 show significant differences in model performance across metrics. The balanced dataset offered an accuracy of 85.6%, precision of 89.7%, Recall of 80.0%, and an F1 score of 84.8%, with a ROC-AUC of 93.3%, when applied to the 25% dataset. However, when the dataset was unbalanced, accuracy dropped slightly to 83.6%; precision fell to 74.79%; recall increased to 89%; F1-score fell to 81.28%; and ROC-AUC remained the same at 88.99%. For the 50% dataset, the balanced dataset showed higher overall accuracy, achieving 86.5% accuracy, 91.57% precision, 80.4% recall, 85.62% F1-score, and 94.46% ROC-AUC. In the case of an unbalanced dataset, there is a significant decrease in accuracy to 68.8%, a considerably higher ROC-AUC of 93.98%, an imbalanced precision of 95.57%, reduced Recall to 50.33%, and a much lower F1 score of 65.94%. With the 75% dataset, the balanced dataset maintained better performance with an accuracy of 86.47%, precision of 92.01%, recall of 79.87%, F1-score of 85.51%, and ROC-AUC of 94.87%.

TABLE 4: Results Based on Performance Metrics

Dataset	Balanced					Unbalanced				
Metrics	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**ROC-AUC**	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**ROC-AUC**
25%	85.60%	89.73%	80.04%	84.80%	93.30%	83.60%	74.79%	89.00%	81.28%	88.99%
50%	86.50%	91.57%	80.40%	85.62%	94.46%	68.80%	95.57%	50.33%	65.94%	93.98%
75%	86.47%	92.01%	79.87%	85.51%	94.87%	86.13%	79.70%	87.67%	83.49%	93.71%
100%	87.80%	92.00%	82.80%	87.16%	95.33%	74.50%	95.04%	60.67%	74.06%	93.91%

On the other hand, in an unbalanced dataset, an accuracy of 86.13%, precision of 79.7%, Recall of 87.67%, F1-Score of 83.49%, and ROC-AUC of 93.71% were obtained, indicating improvement over the results for the 50% unbalanced dataset. Lastly, considering the 100% dataset, the accuracy of the balanced dataset was 87.8%, precision 92%, recall 82.8%, F1 score 87.16%, and ROC-AUC 95.33%, indicating that the model outperformed in this combination. However, the unbalanced dataset performs poorly, achieving an accuracy of 74.5%, precision of 95.04%, Recall of 60.67%, F1-score of 74.06%, and a ROC-AUC of 93.91%. The graphs in Figure 8 show results for different metrics using different combinations of balanced and unbalanced datasets.

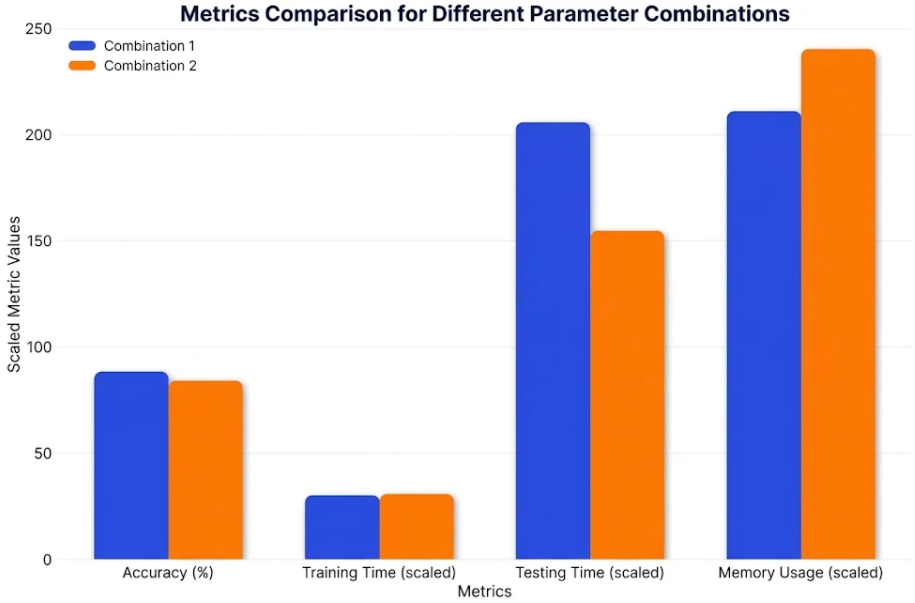


FIGURE 7: Comparison of Two Combination Settings Based on Different Performance Metrics

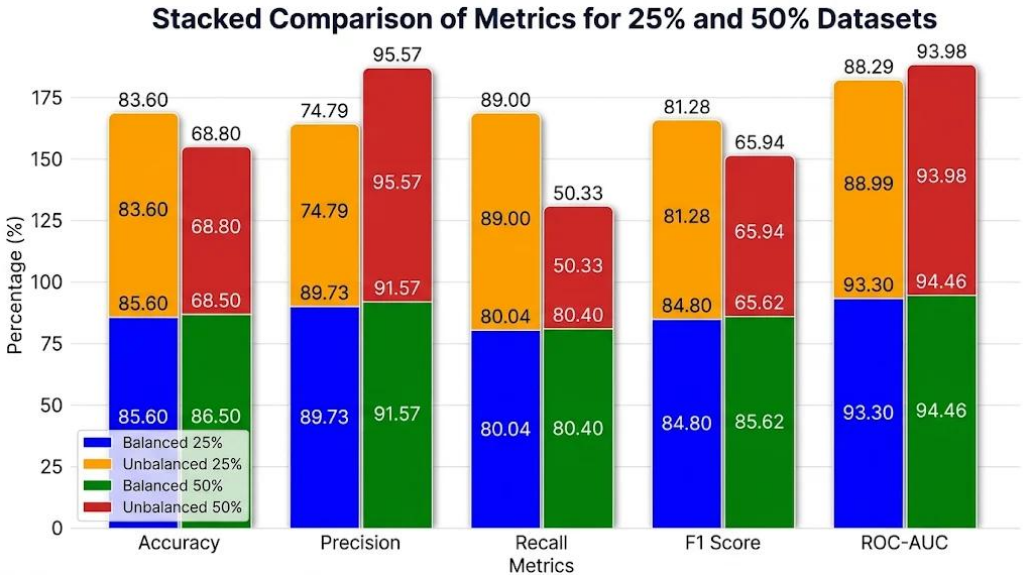


FIGURE 8: Comparison of 25% and 50% of Balanced and Unbalanced Data

B. Results Based on Resource Usage

On the 25% dataset, the balanced dataset had a training time of 373.01 seconds and a test time of 7.59 seconds, for a total time complexity of 380.97 seconds, with a memory consumption of 2.536 GB. However, the unbalanced dataset results in a training time of 303.72 s, a testing time of 6.77 s, a memory usage of 2.64 GB, and a total time complexity of 310.48 s. For the 50% combination, the balanced dataset took 658.93 seconds (638.09 seconds for training and 20.84 seconds for testing) to process, with a memory usage of 2.69 GB. For an unbalanced dataset, training takes 406.04 seconds, and testing takes 8.39 seconds, for a total time complexity of 414.44 seconds, with a slight decrease in memory usage to 2.52 GB. On the balanced dataset, training on 75% of the data took 935.99 seconds, and testing took 14.95 seconds, resulting in a total time complexity of 950.93 seconds and a memory usage of 2.62 GB.

On the unbalanced dataset, the training and testing times were 748.61 seconds and 11.14 seconds, respectively, yielding an overall time complexity of 759.75 seconds, and the lowest observed memory usage was 2.25 GB across all configurations. Lastly, when the dataset was balanced, it yielded the highest training time of 1329.59s, a testing time of 20.82 s, a total time complexity of 1350.41 s, and a memory usage of 2.54 GB. On the other hand, the unbalanced dataset has a much lower training time of 791.20 s and a testing time of 20.83 s, resulting in a total time complexity of 812.03 s and memory usage of 2.57 GB. The comparison graph of memory usage and time complexity is shown as follows in Figure 9.

TABLE 5: Results Based on Resource Usage

Dataset Combination	Training Time (s)	Testing Time (s)	Total time (s)	Memory (GB)
25% – Balanced	373.01	7.59	380.97	2.536
25% – Unbalanced	303.72	6.77	310.48	2.64
50% – Balanced	638.09	20.84	658.93	2.69
50% – Unbalanced	406.04	8.39	414.44	2.52
75% – Balanced	935.99	14.95	950.93	2.62
75% – Unbalanced	748.61	11.14	759.75	2.25
100% – Balanced	1329.59	20.82	1350.41	2.54
100% – Unbalanced	791.20	20.83	812.03	2.57

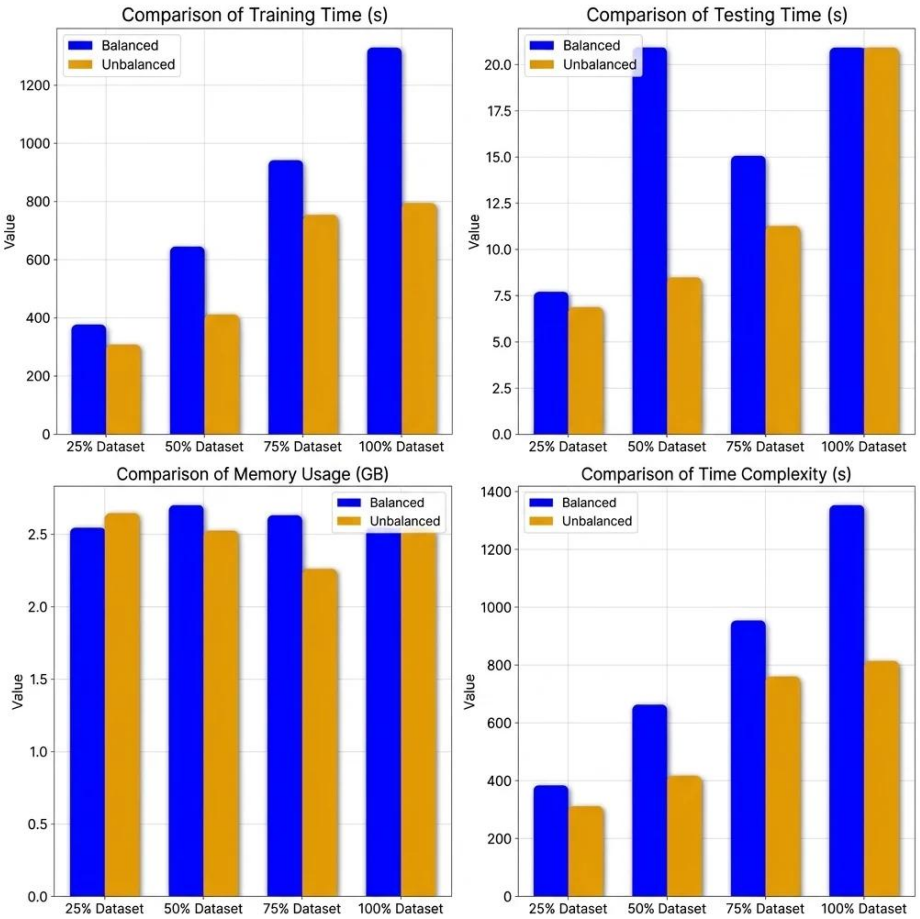


FIGURE 9: Comparison of Training Time, Testing Time, Memory Usage and Time Complexity

C. *Experimentation using Different Ratios of Training and Testing*

For this experiment, different dataset ratios were used to analyze the model's performance. Table 6 presents the distribution of images in each class (Benign and Malignant) across different ratios (90:10, 80:20, 70:30, 60:40, and 50:50).

D. *Results of Different Training-Testing Ratios*

Based on Table 7, for 50%-50%, 60%-40%, 70%-30%, 80%-20%, and 90%-10% splits, the 80%-20% split proved to be the best across all metrics such as accuracy (86.31%), precision (86.76%), Recall (85.35%), and ROC-AUC (93.88%). It is observed that the 70%-30% split yielded the minimum training time (1149.96s) and time complexity (1191.14s), while the 90%-10% split had the maximum Recall (85.58%). The lowest memory usage was observed for the 70%-30% split (2.27GB), and the testing time was lowest for the 90%-10% split (13.03s).

TABLE 6: Dataset Distribution

Ratio	Train Dataset	Test Dataset
Training 50%, Testing 50%	Malignant: 3295 images Benign: 3644 images	Malignant: 3295 images Benign: 3645 images
Training 60%, Testing 40%	Malignant: 3954 images Benign: 4373 images	Malignant: 2636 images Benign: 2916 images
Training 70%, Testing 30%	Malignant: 4613 images Benign: 5102 images	Malignant: 1977 images Benign: 2187 images
Training 80%, Testing 20%	Malignant: 5272 images Benign: 5831 images	Malignant: 1318 images Benign: 1458 images
Training 90%, Testing 10%	Malignant: 5931 images Benign: 6560 images	Malignant: 659 images Benign: 729 images

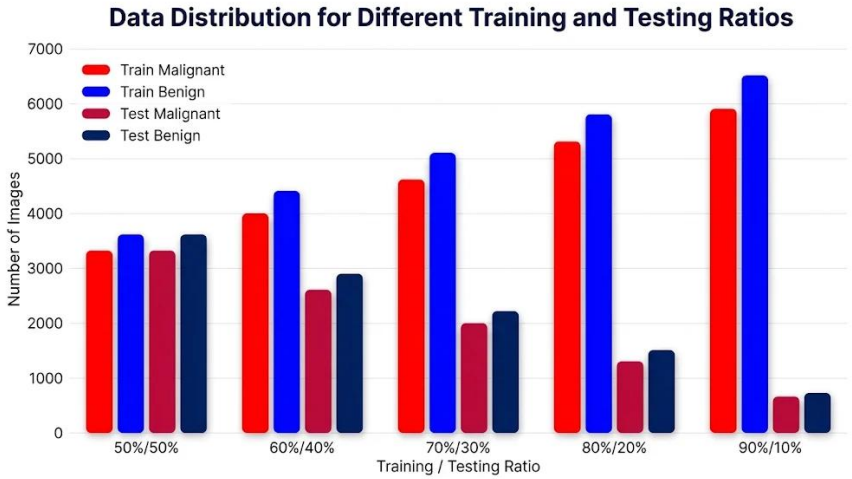


FIGURE 10: Distribution of the Training and Test Datasets Across Different Training-Testing Ratios

VI. DISCUSSION

A. *Analysis of Results Based on Balanced and Unbalanced Dataset*

In this, results are analyzed using two sets of metrics: performance metrics such as Accuracy, Precision, Recall, F1-score, and ROC-AUC. Resource utilization metrics include training time, testing time, memory usage, and time complexity.

TABLE 7: Metrics Evaluation Across Different Training-Testing Ratios

Metrics	Training 50%, Testing 50%	Training 60%, Testing 40%	Training 70%, Testing 30%	Training 80%, Testing 20%	Training 90%, Testing 10%
Accuracy	84.78%	85.36%	85.76%	86.31%	85.95%
Precision	85.24%	86.95%	86.50%	86.76%	84.94%
Recall	82.19%	81.37%	82.95%	83.99%	85.58%
F1 Score	83.68%	84.07%	84.69%	85.35%	85.26%
Support	6940	5552	4164	2776	1388
ROC-AUC	92.49%	93.08%	93.44%	93.88%	93.43%

Training Time	1244.97s	1304.31s	1149.96s	1339.47s	1228.83s
Testing Time	49.67s	40.56s	41.18s	26.61s	13.03s
Memory Usage	2.96GB	2.68GB	2.27GB	2.65GB	2.67GB

B. Analysis of Results Based on Performance Metrics

For the balanced dataset, as dataset size increases, the model's overall performance improves in terms of accuracy, precision, Recall, F1 score, and ROC-AUC. The balanced dataset yields better results because each class has the same number of images, making it easier for the model to learn to detect both benign and malignant cases. On the other hand, there are fluctuations in an unbalanced dataset in terms of accuracy, such as a decrease to 50%, followed by increases to 75% and 100%. This indicates that larger unbalanced datasets do not always yield better results. For the unbalanced dataset, precision reaches 100% because the minority class is predicted correctly on any given sample. Recall, which measures the ability to predict the minority class correctly, drops from 100% to 50%, then improves to 75% but remains lower than in the balanced dataset. In Figure 12, F1 score exhibits behavior similar to Recall and precision, decreasing to 50% and increasing with larger datasets. Increasing the size of the unbalanced dataset improves ROC-AUC, but it remains lower than that of the balanced dataset, indicating poorer class discrimination.

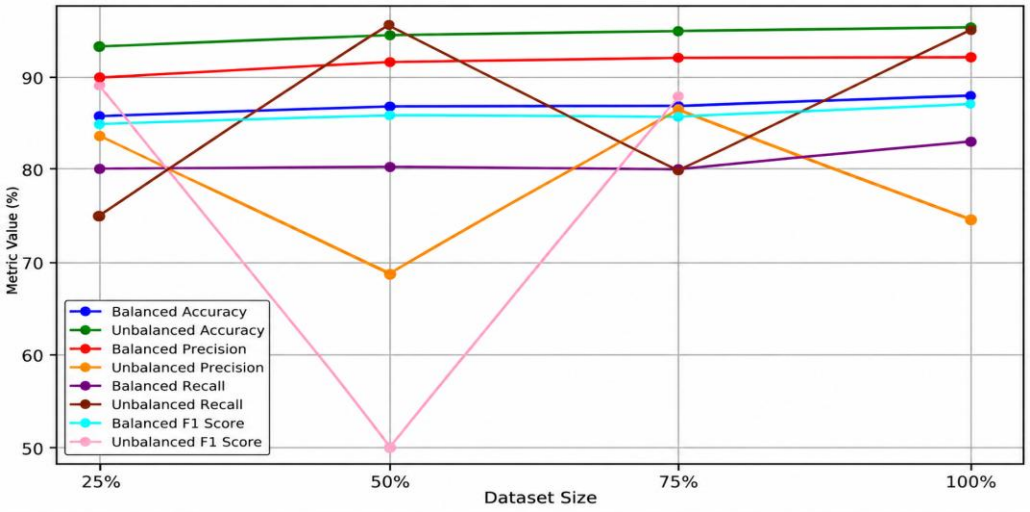


FIGURE 11: Comparison of Balanced and Unbalanced Datasets

In all metrics covered, accuracy, F1 score, and ROC AUC, the balanced dataset outperforms the unbalanced one for all dataset sizes, indicating that the balanced dataset is superior in classifying both classes. Although an unbalanced dataset sometimes performs better on Recall or precision, these benefits often come with biases toward the majority class, hindering generalization and model fairness. These results emphasize necessary steps to address class imbalance to obtain more accurate and more reliable models. The following graphs in Figures 13 show a comparison of metrics across different dataset combinations, i.e., balanced and unbalanced.

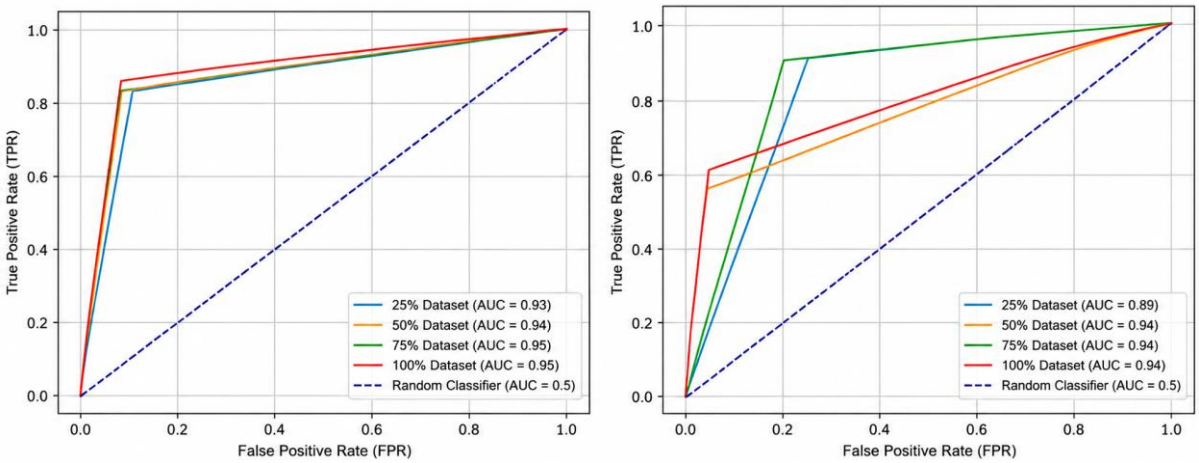


FIGURE 13: ROC-AUC Curve of Balanced and Unbalanced Dataset

C. Analysis of Results Based on Resource Usage

The outcomes reveal a trade-off between computational resources and model performance. The requirement to use balanced datasets necessitates increased computational resources to improve efficiency, resulting in higher accuracy,

Recall, and F1 scores. There is a positive correlation between larger datasets, higher time complexities, and longer training periods for models trained on such datasets, resulting in increased memory allocation requirements compared to models trained on unbalanced data. Figure 11 shows that balanced datasets consumed up to 2.69 GB and 2.54 GB of memory for the 50% and 100% cases, respectively. This high consumption of computational resources during the training and testing phases reduces the efficiency of balanced datasets.

On the contrary, the use of unbalanced datasets results in higher efficiency through less training time and lower memory utilization requirements. Training took 406.04 seconds for the 50% unbalanced dataset and 638.09 seconds for the same proportion of balanced datasets. Unbalanced datasets are computationally efficient, making it easier and faster to conduct experiments under limited computational resources. Unfortunately, unbalanced data suffers from several drawbacks. First, their intrinsic characteristics limit the ability to learn from data effectively. For instance, training on unbalanced data is prone to majority-class overfitting, thus resulting in poor Recall and F1 scores for minority classes. Second, balanced data are relatively stable across learning curves and yield better classification outcomes with fewer errors, as indicated by accuracy plots, confusion matrices, and ROC curves. However, achieving such desirable outcomes comes at the expense of increased computational costs.

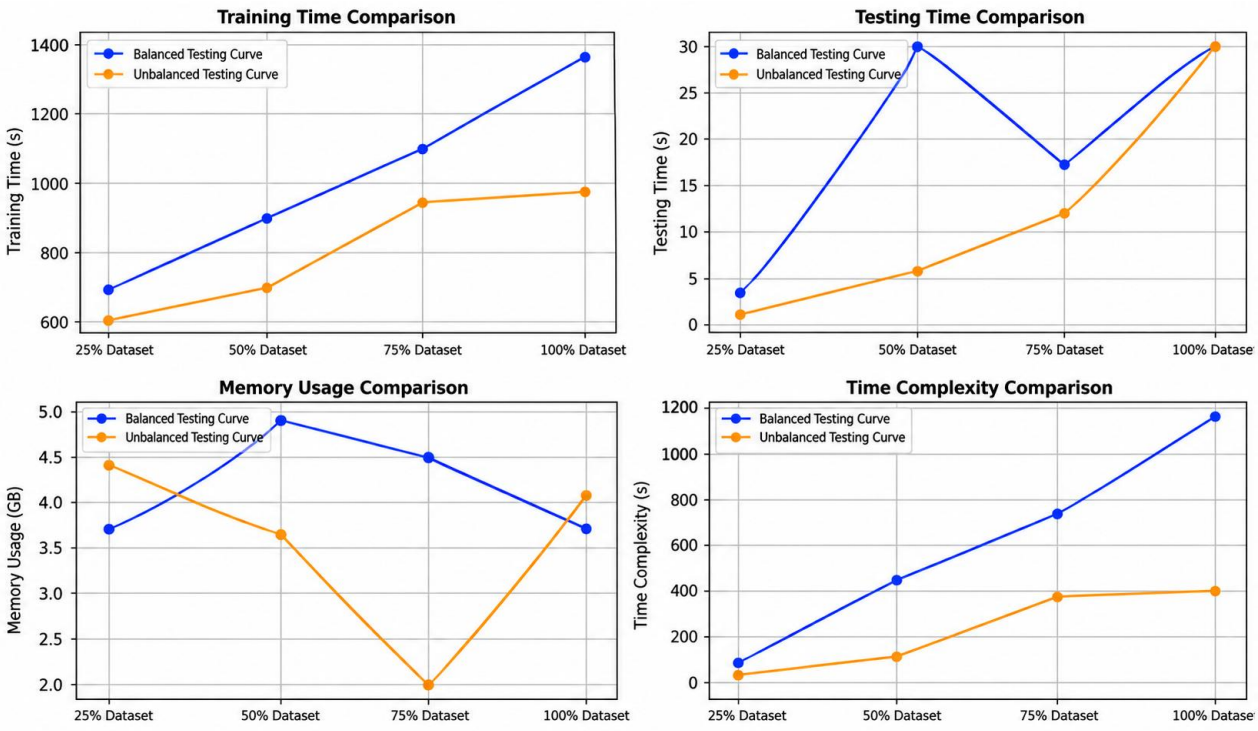


FIGURE 12: Comparisons of Performance Metrics

D. Analysis of Results Based on Resource Usage

Performance assessments using varying training-to-testing partitions (50-50%, 60-40%, 70-30%, 80-20%, and 90-10%) yielded important observations regarding model performance and computational resource demands. Model performance assessed via accuracy, precision, F1 score, and area under the curve (AUC) increased gradually with an increase in training data (from 50% to 90%). On the other hand, training costs increase significantly with increasing training data. For example, training time reached a peak of 1339.47 seconds for the 80-20% partition, whereas test time and memory requirements showed more modest variations. Particularly, test time decreased marginally with higher training data proportions, achieving the shortest time of 13.03 seconds in the 90-10% partition.

Memory requirements showed a similar pattern, decreasing from 2.96 GB in the 50-50% partition to 2.27 GB in the 70-30% partition, then leveling off at 2.65 GB. Similarly, time complexity mirrored the trend observed in training times, with the highest value achieved by the 80-20% partition. Overall, these outcomes demonstrate a clear trade-off between efficiency and performance gains. An expansion of the training dataset results in improvements in accuracy, precision, Recall, F1-score, and AUC. However, there is a parallel rise in training times and time complexity. Therefore, choosing the right training-to-test split is crucial to achieving the required balance. According to the empirical results, a training-testing ratio of 80-20% yields satisfactory performance while minimizing computational resources used. Figure 13 illustrates the performance measures according to each split proportion.

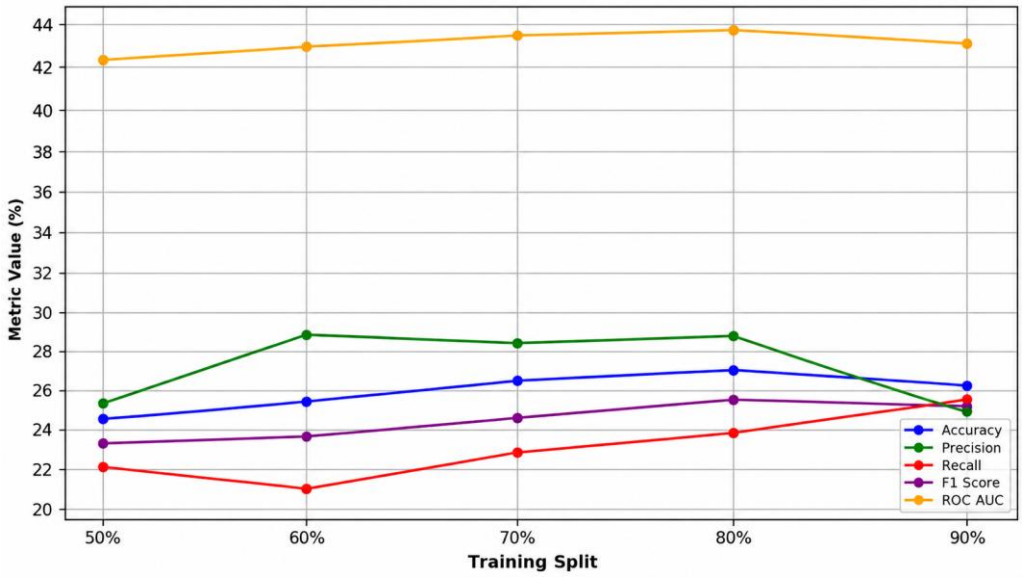


FIGURE 13: Performance Metrics of Training-Testing Ratios

VII. CONCLUSION

This study experimentally evaluates the ability of the Xception model to classify melanoma skin cancers, exploring how dataset distributions, dataset proportions, training-testing splits, and model parameter settings may influence performance and computational requirements. Findings show consistently better performance in terms of accuracy, Recall, F1 score, and ROC-AUC for balanced datasets, compared with unbalanced ones. Even though precision values were higher for unbalanced datasets, these datasets had much lower Recall and F1 scores, indicating difficulty in identifying minority classes during melanoma classification. Combination 1 of the hyperparameters (learning rate=0.001, batch size=32) showed the best performance among the combinations considered, despite significant variation across combinations in dataset distributions and train-test ratios. There was an improvement in classification performance with larger datasets, especially balanced ones. Testing different splits indicated that 80% training data and 20% testing data achieved optimal performance, with accuracy, F1 score, and ROC-AUC of 86.31%, 85.35%, and 93.88%, respectively. Increased dataset sizes resulted in higher computational costs, including extended training durations and increased storage requirements.

DECLARATIONS

Author Contributions: All authors contributed equally to this work.

Funding: The authors received no external funding for this research.

Competing Interests: The authors declare no competing interests.

Data Availability: The datasets analyzed in this study are publicly available from their respective sources. Further information regarding dataset sources and access is provided in the manuscript.

REFERENCES

[1] H. Mahmud et al., "An interpretable deep learning approach for skin cancer categorization," Diagnostics, vol. 13, no. 18, Art. no. 3012, 2023.

[2] M. A. Raza et al., "Optimizing skin cancer screening with convolutional neural networks in smart healthcare systems," PLOS ONE, vol. 20, no. 1, Art. no. e0317181, 2025.

[3] M. Bouzidi et al., "Towards explainable skin cancer diagnosis: A Vision Transformer approach with Grad-CAM visualization," Journal of Imaging, vol. 11, no. 3, Art. no. 85, 2025.

[4] S. Kollem et al., "SKIN-MARS: Multi-scale attention-driven residual Swin-Transformer for explainable skin cancer classification," Biomedical Signal Processing and Control, vol. 91, Art. no. 106010, 2025.

[5] S. Cherukuri and G. Vara Prasad, "EnsembleSkinNet: A transfer learning-based framework for efficient skin cancer detection with explainable AI integration," Frontiers in Oncology, vol. 15, Art. no. 1699960, 2025.

[6] S. Nusrat et al., "Skin cancer detection: Leveraging hybrid deep learning models and traditional machine learning classifiers," Scientific Reports, vol. 15, no. 1, Art. no. 4120, 2025.

[7] B. Benaly et al., "Explainable AI for skin cancer classification: Unlocking insights with Grad-CAM and Grad-CAM++," Computers in Biology and Medicine, vol. 172, Art. no. 108210, 2025.

[8] M. F. Khan et al., "SynthraXCoreNet: An interpretable, well-calibrated six-CNN ensemble for dermoscopic skin-lesion classification," IEEE Access, vol. 13, pp. 18500–18520, 2025.

[9] N. Malhotra and P. Kaur, "Exploring explainable melanoma classification: Leveraging pre-trained deep learning model on MED-NODE dataset," Computer Methods and Programs in Biomedicine, vol. 245, Art. no. 108010, 2024.

[10] S. T. H. Shah et al., "Explainable AI-based skin cancer detection using CNN, particle swarm optimization and machine learning," Journal of Imaging, vol. 10, no. 12, Art. no. 332, 2024.

[11] A. Abualkebash et al., "Explainable depth-wise and channel-wise fusion models for multi-class skin lesion classification," PLOS ONE, vol. 21, no. 2, Art. no. e0340901, 2026.

[12] A. Augustinne et al., "Addressing real-world domain shifts in skin lesion diagnosis with metric-based few-shot learning," IEEE Access, vol. 13, pp. 24150–24165, 2025.

[13] S. Q. Gilani et al., "Adversarial training-based domain adaptation of skin cancer images," Life, vol. 14, no. 8, Art. no. 1009, 2024.

[14] T. Jaya et al., "Federated learning with explainable AI for robust skin lesion classification in diverse clinical settings," Journal of Healthcare Engineering, 2025, Art. no. 4512980.

[15] A. R. Priambodo and C. Fatichah, "Evaluating lightweight CNN models with CBAM for explainable AI skin cancer classification using GradCAM and ScoreCAM," IEEE Access, vol. 13, pp. 29810–29822, 2025.

[16] D. Corbin et al., "Assessing bias in skin lesion classifiers with contemporary deep learning and post-hoc explainability techniques," IEEE Access, vol. 11, pp. 134210–134225, 2023.

[17] N. Sultana et al., "Domain adaptive skin lesion classification via conformal ensemble of Vision Transformers," IEEE Transactions on Artificial Intelligence, vol. 6, no. 2, pp. 340–352, 2025.

[18] J. Ye, "Enhancing skin lesion classification generalization with active domain adaptation," Pattern Recognition Letters, vol. 178, pp. 115–122, 2025.

[19] Y. Jadhav et al., "Enhancing skin cancer diagnosis with XAI: Integrating Grad-CAM, SHAP, and LIME for model interpretability," IEEE Transactions on Medical Imaging, vol. 44, no. 4, pp. 1120–1132, 2025.

[20] F. Lubis et al., "ReXNet-150 and Grad-CAM for efficient and explainable melanoma skin cancer detection," Journal of King Saud University - Computer and Information Sciences, vol. 37, no. 1, Art. no. 102230, 2025.

[21] R. Vinitha et al., "A robust and interpretable deep learning-based system for multi-category skin cancer categorization on resource-constrained edge devices," Microprocessors and Microsystems, vol. 105, Art. no. 104980, 2025.