# Performance Analysis of MobileNet: Effects of Dataset Size, Class Balancing, and Data Split Ratios

Zainab Asim[*], Mujeeb-Ur-Rehman [†]

***Abstract***

*Image classification plays an important role in computer vision and has a variety of real-world applications. In order to perform an image classification task efficiently, among the most popular lightweight Convolutional Neural Network models is the MobileNet, which is optimized to be utilized in resource-constrained devices. In this paper, we assess the binary image classification task using the MobileNet architecture, with consideration of the three experimental factors, including the effects of dataset size, dataset balance and the train-test split ratio. We tested the MobileNet model with different size ratios of the Cats and Dogs dataset, such as 25%, 50%, 75%, and 100% with balanced and unbalanced data partitions. The findings indicate that the size and balance of the dataset have a significant impact on the classification accuracy, and results on the balanced datasets consistently outperform the unbalanced datasets. In addition, the MobileNet model performed well in the experiment with multiple train-test splits, including ratios of 50%:50%, 60%:40%, 70%:30%, 80%:20%, and 90%:10%, especially when using a 70% training and 30% testing split.*

***Keywords:*** MobileNet, Image Classification, Deep Learning, Dataset Balancing, Computer Vision.

## Introduction

Image classification has become an essential challenge in computer vision, with applications in diverse domains, including in agriculture (Akbar et al., 2024), healthcare (Serte et al., 2022), and e-commerce (Rui, 2020). The classic Machine Learning (ML) algorithms were primarily based on features that were handcrafted, and they could not be generally resilient to different data distributions. However, advancements in Deep Learning, especially Convolutional Neural Networks (CNNs), have revolutionized image classification tasks by providing an automated extraction of features and increasing the generalization of the models. In spite of these advancements, it is still challenging to deploy high performance CNNs on hardware with limited processing capabilities. To address this problem, a lightweight MobileNet model was proposed (Howard et al., 2017). The lightweight structure of the MobileNet model alongside the depth-wise separable convolution usage, covers this issue through a significant reduction in the cost of

[*]Department of Computer Science, University of Management and Technology, Sialkot 51310, Pakistan, zainabasim211@gmail.com

[†]Corresponding Author: Department of Computer Science, University of Management and Technology, Sialkot 51310, Pakistan, mujeeb.rehman@skt.umt.edu.pk

computation and memory consumption without compromising the accuracy.

While the architecture of the MobileNet is fundamentally efficient, its performance can be modified by many aspects, such as the hyperparameter settings, size of the dataset, its balance, and different train-test split ratios. Hyperparameters such as the number of epochs, batch size, and learning rate have a significant impact in determining the convergence rate and overall performance of the model (Wojciuk et al., 2024). Therefore, an improper choice of hyperparameters might lead to suboptimal learning or overfitting issues. Likewise, dataset imbalance is also a common challenge in supervised learning. The models trained on imbalanced datasets often exhibit bias toward the majority classes, thereby compromising the detection of minority class instances (Ali et al., 2013). However, methods like oversampling, under-sampling, and synthetic data generation techniques such as SMOTE have been introduced to alleviate this issue to some extent (Chawla et al., 2002). Moreover, the size of the datasets also affects the evaluation of a model's generalization capability (Soekhoe et al., 2016). In addition, the model's performance is greatly impacted by the proportions of training and testing data (Bichri et al., 2024). An overly skewed split can either lead to insufficient training or unreliable performance evaluation.

Although several studies have examined the individual effects of the size of datasets (Soekhoe et al., 2016), the balancing of datasets (Gosain & Sardana, 2017), and the effect of the train-test splits (Bichri et al., 2024) on classification models, the comprehensive investigations that analyze their combined impact with the lightweight models such as MobileNet are scarce. Thus, this research aims to fill this gap by performing a systematic analysis of how these three factors interact to influence MobileNet's classification performance on the widely used Cats and Dogs dataset. The motivations of this study are as follows: (1) to provide insights into the impact of dataset size, (2) to examine how balancing data affects classification outcomes, and (3) to identify the optimal data partitioning ratio for maximizing generalization performance. Our contributions include:

- An extensive evaluation of the MobileNet model's performance under varying dataset sizes.
- A detailed investigation into how dataset distributions (balanced and unbalanced) affect the key performance metrics.
- An in-depth analysis of different train-test split ratios (50%:50%, 60%:40%, 70%:30%, 80%:20%, and 90%:10%) and their effect on model generalization.

## Literature Review

Lightweight neural network architectures have gained a lot of popularity over the last few years, because they can be used in resource-constrained and real-time systems. To obtain the best tradeoff between accuracy and efficiency, the MobileNet architecture was introduced that utilized depth-wise separable convolutions (Howard et al., 2017). This model was more optimized for low processing power and memory devices. Expanding on this, MobileNetV2 came up with added inverted residual blocks and linear bottlenecks to enhance representational power without affecting the computational efficiency (Sandler et al., 2018).

Despite the popularity of MobileNet, limited attention has been paid to systematically analyze the impact of varying dataset sizes, data distributions (balanced and unbalanced), and train-test split ratios on the classification performance. Some studies highlight how important the size of a dataset is for deep learning. Large datasets often lead to higher generalization but with diminishing returns after a certain point (Sun et al., 2017).

In addition, class imbalance is a critical supervised learning issue. Learning from biased data will result in biased models. The model becomes biased against the majority classes, hence reducing the performance of the minority classes (Kaur et al., 2019). A study revealed that imbalanced training data can have a serious impact on the predictions made by the model (Chawla et al., 2002). The effects of the imbalance of the data on Convolutional Neural Networks were also investigated, and several methods of eliminating the class imbalance were compared (Buda et al., 2018). According to the survey (Spelmen and Porkodi, 2018), the SMOTE approach appears to be an appropriate way of overcoming the issue of class imbalance. Furthermore, methods such as oversampling and under-sampling can also be applied to overcome this difficulty (Shelke et al., 2017).

The performance of the model is also significantly impacted by the train-test split ratios. For example, some train-test split ratios (90%:10%, 80%:20%, 70%:30%, and 60%:40%) comparison was made to determine their influence on the performance of three most used pre-trained models (Bichri et al., 2024). In addition, the study (Muraina, 2022) compared how the different training-testing splits influence model performance and concluded that the best split ratio significantly relies on the datasets' sizes, which directly impact model performance.

Although these studies provide valuable insights into individual aspects of model optimization, there remains a lack of systematic analysis combining varying dataset sizes, dataset balancing, and training-testing splitting of data. This research aims to evaluate MobileNet performance

on the Cats and Dogs dataset across different dataset sizes, class balancing, and train-test split ratios, providing useful insights for future researchers and practitioners.

## Methodology

In this section, the methodology that is used to analyze the performance of the MobileNet model across varying dataset sizes, class-balancing, and training-testing splits is discussed. The adopted methodology is also elaborated in Figure 1 from the dataset collection to the model training and evaluation stage.

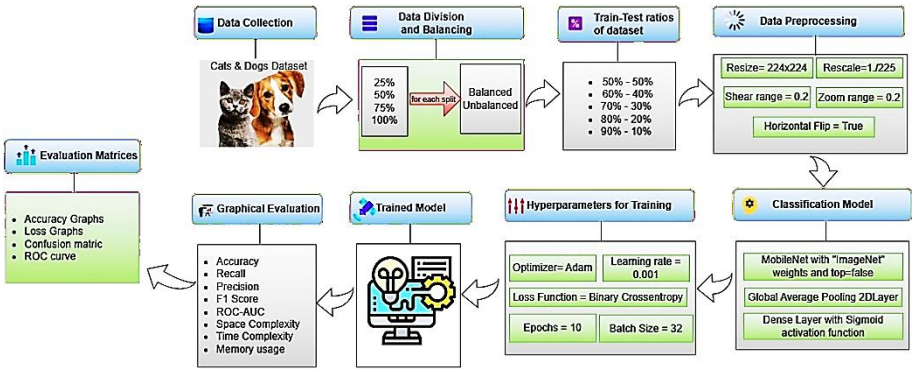


***Figure 1: Methodology Diagram.***

### *Dataset Collection*

In this study, the "cats and dogs" dataset was collected from Kaggle to perform the binary classification task using the MobileNet model. The images in this dataset are in RGB format, and the dataset size is 68.53MB. This cats and dogs' dataset has three directories: train, test, and validation. Each directory has two further subdirectories named cats and dogs. The train directory collectively has 1999 images for training, containing the cats' subdirectory with 999 images and the dogs' subdirectory with 1000 images. Similarly, the validation directory collectively has 602 images to validate, with 301 images of cats and 301 images of dogs. Further, the test directory of this dataset has a total of 398 images for testing, where the cats' subdirectory has 199 images and the dogs' subdirectory has 199 images.

After the collection of the dataset, it was split into 4 combinations, including 25%, 50%, 75%, and 100% data from it for analysis. Each data split was used with balanced as well as with unbalanced distributions. We also made 5 combinations of train-test splits, such as 50%-50%, 60%:40%, 70%:30%, 80%:20%, and 90%:10% to train and test the MobileNet model. For these splits, we merged the original splits of the dataset and reused

them according to our experimental design to systematically evaluate different train-test ratios.

### *Specifications of Balanced and Unbalanced Datasets*

Firstly, 25% images from the cats and dogs' dataset were used for analysis. After that, we kept on expanding the dataset to 50%, 75%, and then 100%. For each proportion, we made a balanced and an unbalanced dataset. In balanced datasets, the number of images in both classes was the same, while in unbalanced datasets, the cats and dogs' dataset had images of a 2:1 ratio, whereas the majority class (the cats' class) had twice as the images as the minority class (dog class). The number of images in each balanced and unbalanced dataset is shown in Table 1.

***Table 1: Specifications of the Balanced and Unbalanced datasets of cats and dogs across different divisions.***

| Division (%) | Type | Train | Validation | Test | Images/ class | Total Images |
|---|---|---|---|---|---|---|
| 25 | Balanced | Cats: 250<br>Dogs: 250 | Cats: 76<br>Dogs: 76 | Cats: 50<br>Dogs: 50 | Cats: 376<br>Dogs: 376 | 752 |
| | Unbalanced | Cats: 250<br>Dogs: 125 | Cats: 76<br>Dogs: 38 | Cats: 50<br>Dogs: 25 | Cats: 376<br>Dogs: 188 | 564 |
| 50 | Balanced | Cats: 500<br>Dogs: 500 | Cats: 151<br>Dogs: 151 | Cats: 100<br>Dogs: 100 | Cats: 751<br>Dogs: 751 | 1502 |
| | Unbalanced | Cats: 500<br>Dogs: 250 | Cats: 151<br>Dogs: 76 | Cats: 100<br>Dogs: 50 | Cats: 751<br>Dogs: 376 | 1127 |
| 75 | Balanced | Cats: 750<br>Dogs: 750 | Cats: 226<br>Dogs: 226 | Cats: 150<br>Dogs: 150 | Cats: 1126<br>Dogs: 1126 | 2252 |
| | Unbalanced | Cats: 750<br>Dogs: 375 | Cats: 226<br>Dogs: 113 | Cats: 150<br>Dogs: 75 | Cats: 1126<br>Dogs: 563 | 1689 |
| 100 | Balanced | Cats: 999<br>Dogs: 999 | Cats: 301<br>Dogs: 301 | Cats: 199<br>Dogs: 199 | Cats: 1499<br>Dogs: 1499 | 2998 |
| | Unbalanced | Cats: 999<br>Dogs: 500 | Cats: 301<br>Dogs: 151 | Cats: 199<br>Dogs: 100 | Cats: 1499<br>Dogs: 751 | 2250 |

### *Data Preprocessing*

The preprocessing of data was done by using the Keras library, which involved different techniques such as resizing, rescaling, and image augmentation, including shear, zoom, and horizontal flip.

### *Model Architecture*

To perform classification, the MobileNet model with "ImageNet" weights was taken using the Keras library, followed by two newly added layers Global Average Pooling 2D layer and a Dense layer with sigmoid activation function for a binary classification task.

### *Model Training and Evaluation*

The model was set for training by adjusting hyperparameters such as the use of the Adam optimizer, learning rate = 0.001, and the binary

cross-entropy loss function. Furthermore, the model was trained with 10 epochs and 32 batch size, as shown in Table 2. Moreover, to evaluate the performance of the trained model, different evaluation matrices were considered, such as accuracy, precision, recall, F1-score, ROC-AUC, time complexity, space complexity, and memory usage.

***Table 2: List of hyperparameters along with purpose and values used in the experimentation of MobileNet***

| Hyperparameters | Purpose | Values |
|---|---|---|
| Optimizer | It highlights the algorithm used for the optimization. | Adam |
| Learning Rate | It controls the step size with which the weights of the model are updated during training. | 0.001 |
| Loss Function | It calculates the error between the actual and predicted outputs to improve the learning of the model. | Binary Cross-Entropy |
| Data Augmentation | It helps in applying different transformations to improve the diversity of data. | Rescale: 1./255<br>Shear range: 0.2,<br>Zoom range: 0.2,<br>Horizontal Flip: True |
| Activation Function | It helps the model to learn complex patterns by introducing non-linearity. | Sigmoid |
| Epochs | It specifies the number of times the training data is passed through the model. | 10 |
| Batch Size | It is the number of samples that have to be processed in a batch. | 32 |

## Results and Analysis

### B*alanced and Unbalanced Datasets Experiment across different Data Divisions*

In this experiment, the evaluation of the model's performance was done on both balanced and unbalanced datasets using different train-test split ratios, such as 25%, 50%, 75%, and 100% as shown in Table 3. The results indicated that when using a 25% balanced dataset, the model achieved an accuracy of 96%, with a precision of 97.92%, recall of 94.00%, an F1-score of 95.92%, and an ROC-AUC of 99.76%. In contrast, on the 25% unbalanced dataset, the model's performance was degraded than the 25% balanced dataset, showing 94.67% accuracy, 95.65% precision, 88.00% recall, 91.67% F1-score, and 99.36% of ROC score. Similarly, when the model was executed on 50% balanced and unbalanced datasets, it showed a 98.00% accuracy score, a precision of 97.06%, a recall of 99.00%, an F1-score of 98.02%, and a 99.89% ROC-score with the 50% balanced dataset. On the 50% unbalanced dataset, an accuracy rate of 97.33%, precision, recall, and F1-score of 96%, and ROC-AUC of 99.76% were obtained.

The model was also trained and tested on 75% of both balanced and unbalanced datasets. It provided accuracy, precision, recall, F1-score, and ROC-AUC for a balanced 75% dataset were 98.33%, 97.39%,

99.33%, 98.35%, and 99.92%, respectively. In the case of the unbalanced 75% dataset, the models showed lower scores than balanced, such as an accuracy of 97.78%, a precision of 97.30%, a recall of 96.00%, an F1-score of 96.64%, and an ROC score of 99.77%. On a complete 100% balanced dataset, the model presented an excellent performance in classifying cats and dogs, displaying results including 98.74% accuracy, 98.50% precision, 98.99% recall, 98.75% F1-score, and an ROC-AUC score of 99.86%. In contrast, the 100% unbalanced dataset results were inferior to 100% balanced, showing accuracy, precision, recall, F1-score, and ROC-AUC of 97.66%, 96.97%, 96.00%, 96.48%, and 99.89% respectively.

***Table 3: Performance of modified MobileNet model on both Balanced and Unbalanced datasets using different data splits.***

| Division (%) | | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) | ROC (%) |
|---|---|---|---|---|---|---|
| 25 | Balanced | 96.00 | 97.92 | 94.00 | 95.92 | 99.76 |
| | Unbalanced | 94.67 | 95.65 | 88.00 | 91.67 | 99.36 |
| 50 | Balanced | 98.00 | 97.06 | 99.00 | 98.02 | 99.89 |
| | Unbalanced | 97.33 | 96.00 | 96.00 | 96.00 | 99.76 |
| 75 | Balanced | 98.33 | 97.39 | 99.33 | 98.35 | 99.92 |
| | Unbalanced | 97.78 | 97.30 | 96.00 | 96.64 | 99.77 |
| 100 | Balanced | 98.74 | 98.50 | 98.99 | 98.75 | 99.86 |
| | Unbalanced | 97.66 | 96.97 | 96.00 | 96.48 | 99.89 |

It is observed from these results that the performance of balanced datasets is better than that of unbalanced datasets across different ratios because unbalanced datasets lead to biased classifications, which affects the performance of the model. Also, it is visible that with the increase in the data, the performance of the model improves because the deep learning models are data-hungry. Thus, the MobileNet model achieved significant results on a 100% balanced dataset. The graphical representation of these results is also shown in Figure 2, for better understanding.

### *Balanced and Unbalanced Datasets Resource Usage Experiment across different Data Divisions*

The results obtained when the model was run on an unbalanced and balanced dataset across four proportions of data, considering the metrics such as memory usage, training time, testing time, and total time, are discussed in Table 4. The experiments showed that as the dataset size increased, the model utilized more training and testing time as well as memory. Also, balanced datasets used more resources than unbalanced ones because of the large number of samples they have. The 25% balanced dataset took 255.07 sec to train and 37.83 sec for testing with 1987.56MB

of memory usage, and the 25% unbalanced dataset took 289 sec for training and 70.28 sec for testing with 1954.78MB of memory.

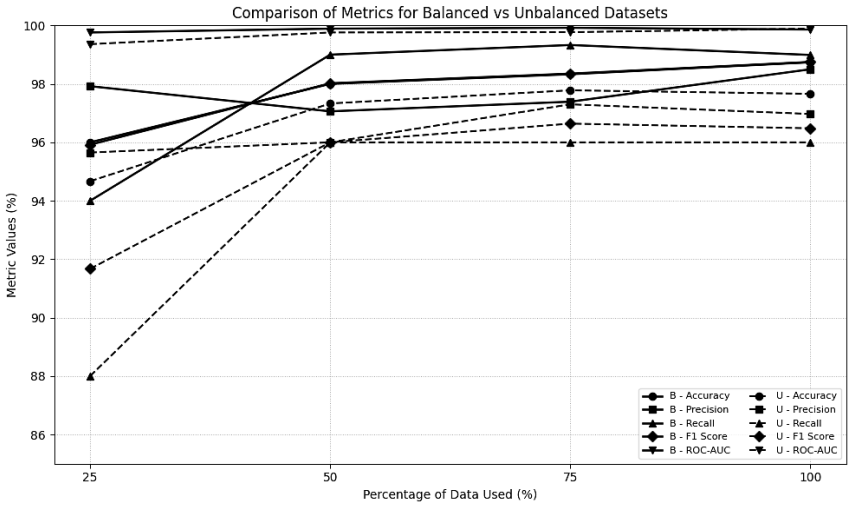


***Figure 2: Performance comparison of balanced and unbalanced datasets across different divisions.***

***Table 4: Resource usage for both Balanced and Unbalanced datasets using different data splits.***

| Division (%) | | Memory usage (MB) | Training Time (sec) | Testing Time (sec) | Total Time (sec) |
|---|---|---|---|---|---|
| 25 | Balanced | 1987.56 | 255.07 | 37.83 | 292.90 |
| | Unbalanced | 1954.78 | 289.00 | 70.28 | 359.28 |
| 50 | Balanced | 2377.66 | 384.74 | 57.82 | 442.56 |
| | Unbalanced | 1976.11 | 305.99 | 42.85 | 348.85 |
| 75 | Balanced | 2464.33 | 727.85 | 108.50 | 836.35 |
| | Unbalanced | 2358.29 | 537.99 | 57.38 | 595.37 |
| 100 | Balanced | 3544.47 | 872.59 | 119.51 | 992.10 |
| | Unbalanced | 2908.23 | 651.95 | 100.42 | 752.37 |

Likewise, for the 50% balanced dataset, 384.74sec for training, 57.82sec for testing, and 2377.66 MB memory were used, and for the unbalanced 50% dataset, the models took 305.99sec and 42.85sec for training and testing, respectively, with 1976.11MB memory. Similarly, the 75% balanced dataset took 836.35sec total time in training and testing, and memory of 2464.33MB, and the unbalanced 75% dataset utilized 2358.29 MB of memory with a total time of 595.37sec. Also, when the model was trained and tested on a 100% balanced dataset, it took more time and memory than others because of the quantity of data, which was 3544.47MB of memory and a total time of 992.10 sec. In contrast, a 100% unbalanced dataset took less time of 752.37 sec, and memory of

2908.23MB, than a 100% balanced dataset. However, minor variations in resource usage can occur due to system-level factors.

### *Train-Test Splits Experiment*

Experiments were also conducted with different ways of splitting the data for training and testing, such as 50% for training and 50% for testing, 60% training and 40% testing, 70% training and 30% testing, 80% training and 20% testing, and 90% training with 10% for testing. These splits are listed in Table 6. The corresponding number of images used in each split is reported in Table 5. The experimental results indicate that the model achieved its best performance with a 70% training and 30% testing split, where 2,098 images of cats and dogs were used for training and 900 images for testing, as illustrated in Figure 3. Under this configuration, the model achieved the highest performance scores, with 98.78% accuracy, 98.24% precision, 99.33% recall, 98.78% F1-score, and 98.78% ROC-AUC. This superior performance of the model can be attributed to the optimal balance between sufficient training data and effective generalization on the test set at this ratio.

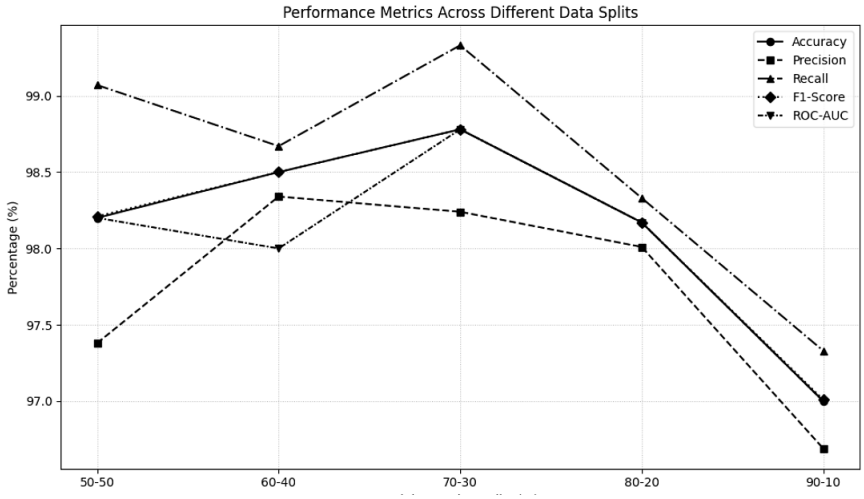


***Figure 3: Comparison of Performance metrics across different training and testing splits.***

***Table 5: Number of Images across different training-testing ratios.***

| Train-Test Ratios (%) | Number of Images |
|---|---|
| 50-50 | Train:1499, Test: 1499 |
| 60-40 | Train:1798, Test: 1200 |
| 70-30 | Train:2098, Test: 900 |
| 80-20 | Train:2398, Test: 600 |
| 90-10 | Train:2698, Test: 300 |

***Table 6: Performance Metrics using different training and testing ratios***

| Train-Test Ratios (%) | Accuracy (%) | Precision (%) | Recall (%) | F1score (%) | ROC (%) |
|---|---|---|---|---|---|
| 50-50 | 98.20 | 97.38 | 99.07 | 98.21 | 98.20 |
| 60-40 | 98.50 | 98.34 | 98.67 | 98.50 | 98.00 |
| 70-30 | 98.78 | 98.24 | 99.33 | 98.78 | 98.78 |
| 80-20 | 98.17 | 98.01 | 98.33 | 98.17 | 98.17 |
| 90-10 | 97.00 | 96.69 | 97.33 | 97.01 | 97.00 |

## Discussion

We evaluated the performance of the modified MobileNet model with the other MobileNet family on the full dataset, as shown in Table 7. This comparison includes MobileNetv2 (Sandler et al., 2018), MobileNetV3Large (Howard et al., 2019), and MobileNetV3Small (Howard et al., 2019). We trained these architectures with the same preprocessing pipeline, hyperparameters, and custom classification head to ensure a fair and consistent evaluation. The results showed that the modified MobileNet achieved the best performance across all evaluation metrics. This indicates that the modified MobileNet improves the model's ability to learn discriminative features while maintaining efficiency.

Although MobileNetV2 achieved comparable results, it was slightly lower than the modified MobileNet. On the other hand, the MobileNetV3Large and MobileNetV3Small models were considerably less efficient in terms of accuracy and precision. This difference can be attributed to differences in the design of each architecture and how well it responds to the selected training settings. These results showed that the modified MobileNet model was more suitable for this dataset and achieved better generalization under the same training conditions. However, this does not necessarily indicate the overall superiority of MobileNet over other variants, since additional tuning of the MobileNetV3 models may improve their results.

***Table 7: Performance Comparison of the models.***

| Models | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) | ROC (%) |
|---|---|---|---|---|---|
| MobileNet | 98.74 | 98.50 | 98.99 | 98.75 | 99.86 |
| MobileNetV2 | 97.24 | 97.00 | 97.49 | 97.24 | 99.65 |
| MobileNetV3Large | 56.03 | 53.90 | 83.42 | 65.48 | 64.40 |
| MobileNetV3Small | 60.05 | 57.19 | 79.90 | 66.67 | 67.13 |

## Conclusion

In this study, we conducted extensive experiments using a pre-trained MobileNet model. The analysis was performed on the Cats and Dogs dataset using four different data proportions (25%, 50%, 75%, and 100%), considering both balanced and imbalanced class distributions. Across all proportions, the model consistently achieved higher performance on balanced datasets compared to imbalanced ones.

Additionally, the performance of the model improved as the size of the training data increased, with the fully balanced dataset (100%) yielding the best overall results. Furthermore, the dataset was evaluated under various training-testing splits, including 50%:50%, 60%:40%, 70%:30%, 80%:205%, and 90%:10%. Among these splits, the 70%:30% split demonstrated superior performance relative to the others. These findings highlight that optimal MobileNet performance depends strongly on appropriate dataset balancing, sufficient data volume, well-selected training-testing ratios, and careful hyperparameter tuning.

**Author Contributions:** Zainab Asim (Conceptualization, Methodology, Experiments, and Writing); Mujeeb-Ur-Rehman (Supervision, Methodology, Editing, and Review)

## References

Akbar, J. U. M., Kamarulzaman, S. F., Muzahid, A. J. M., Rahman, M. A., & Uddin, M. (2024). A comprehensive review on deep learning assisted computer vision techniques for smart greenhouse agriculture. *IEEE Access*, *12*, 4485-4522.

Ali, A., Shamsuddin, S. M., & Ralescu, A. L. (2013). Classification with class imbalance problem. *Int. J. Advance Soft Compu. Appl*, *5*(3), 176-204.

Bichri, H., Chergui, A., & Hain, M. (2024). Investigating the Impact of Train/Test Split Ratio on the Performance of Pre-Trained Models with Custom Datasets. *International Journal of Advanced Computer Science & Applications*, *15*(2).

Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural networks*, *106*, 249-259.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321-357.

Gosain, A., & Sardana, S. (2017, September). Handling class imbalance problem using oversampling techniques: A review. In *2017 international conference on advances in computing, communications and informatics (ICACCI)* (pp. 79-85).

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B., Tan, M., ... & Adam, H. (2019). Searching for mobilenetv3. *In Proceedings of*

*the IEEE/CVF international conference on computer vision* (pp. 1314-1324).

Kaur, H., Pannu, H. S., & Malhi, A. K. (2019). A systematic review on imbalanced data challenges in machine learning: Applications and solutions. *ACM computing surveys (CSUR)*, *52*(4), 1-36.

Muraina, I. (2022, February). Ideal dataset splitting ratios in machine learning algorithms: general concerns for data scientists and data analysts. In *7th international Mardin Artuklu scientific research conference* (pp. 496-504).

Rui, C. (2020). Research on classification of cross-border E-commerce products based on image recognition and deep learning. *IEEE Access*, *9*, 108083-108090.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510-4520).

Serte, S., Serener, A., & Al-Turjman, F. (2022). Deep learning in medical imaging: A brief review. *Transactions on Emerging Telecommunications Technologies*, *33*(10), e4080.

Shelke, M. S., Deshmukh, P. R., & Shandilya, V. K. (2017). A review on imbalanced data handling using undersampling and oversampling technique. *Int. J. Recent Trends Eng. Res*, *3*(4), 444-449.

Soekhoe, D., Van Der Putten, P., & Plaat, A. (2016, September). On the impact of data set size in transfer learning using deep neural networks. *In International symposium on intelligent data analysis* (pp. 50-60).

Spelmen, V. S., & Porkodi, R. (2018, March). A review on handling imbalanced data. In *2018 international conference on current trends towards converging technologies (ICCTCT)* (pp. 1-11).

Sun, C., Shrivastava, A., Singh, S., & Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision* (pp. 843-852).

Wojciuk, M., Swiderska-Chadaj, Z., Siwek, K., & Gertych, A. (2024). Improving classification accuracy of fine-tuned CNN models: Impact of hyperparameter optimization. *Heliyon*, *10*(5).