Performance Analysis of ResNet50 on Balanced and Imbalanced Image Classification Datasets

Kainat Irfan*, Mujeeb-Ur-Rehman †

Abstract

This study explores the performance of a well-known deep learning model ResNet50 on CIFAR-10 and Fashion-MNIST datasets. ResNet50 is an efficient and scalable model for image classification. The model is evaluated on both datasets under balanced and imbalanced conditions. We trained the model using different train-test splits, fine-tuned hyperparameters and applied regularization techniques L1, L2 and dropout. The model is evaluated using Accuracy, Precision, Recall, F1-score, ROC-AUC, Training and Testing time and Memory usage. The results demonstrate that the model performs better on the Fashion-MNIST dataset (86% accuracy) compared to CIFAR-10 dataset (73% accuracy) across all settings. Additionally, the performance of both datasets is better in imbalanced settings but demands more computational resources. These results emphasize how crucial dataset balancing and hyperparameter tuning are to the real-world optimization of deep learning models.

Keywords: ResNet50, Deep Learning, CIFAR-10, Fashion-MNIST, Class Imbalance, Image Classification.

Introduction

Computer vision has been revolutionized with the help of deep learning (O'Mahony, 2020). Because deep learning has automated feature extraction and classification of complex datasets. Among the various architectures, Residual Networks (ResNet) have demonstrated remarkable efficacy in mitigating the vanishing gradient problem in deep neural networks (He, 2016). It's a state-of-the-art deep learning model designed specifically to solve the vanishing gradient problem for deep networks. The aim of this research is to examine the performance of ResNet50 and its popular variant for two datasets: Fashion-MNIST and CIFAR-10. These datasets vary in complexity and image properties, thus providing a comprehensive evaluation of the proposed ResNet50 model.

The model proposes an idea of residual learning, or we can say skip connections, with the help of shortcut connections, which skip layers and pass to one or more layers and enables the model to learn residual mappings well (Hossain, 2022). The aim of this architecture is to train deep neural networks with at least 100 states in the hidden layer without

*Department of Computer Science, University of Management and Technology, Sialkot 51310, Pakistan, kainatirfancs.55@gmail.com

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

sacrificing speed. It also is best known for its scalability, efficiency, and ability to generalize across various tasks such as image classification and object detection as it works on image data (Bello, 2021). The ResNet model is modularly constructed, and its incorporation into diverse deep learning frameworks is also made simple. It also converges much faster due to lower error rates on deep networks and performs very impressively in the computer vision domain, obtaining state-of-the-art results on benchmarks such as ImageNet and CIFAR-10. The transfer learning of the ResNet reduces the need of computationally expensive training as since we extracted salient features and predicted with pre-learned weights.

Class imbalance is also a common issue in real-world applications that can negatively affect the accuracy and reliability of machine-learning models (Buda, 2018). The main objective of this research is to examine how ResNet50 performs under balanced and imbalanced dataset conditions. The contributions of this study are as follows: (a) Evaluating the ResNet50 performance on CIFAR-10 and Fashion-MNIST datasets under balanced and imbalanced settings. (b) Applying different train-test splits to determine the performance of the model. (c) Performing an in-depth analysis on different regularization techniques: L1, L2, and dropout. (d) Performance measurement of the proposed model based on accuracy, Precision, Recall, F1-score, ROC-AUC, Training/Testing time, and Memory usage.

Literature Review

Residual Networks (ResNets) are introduced by He (2016) to address the challenges associated with training very deep neural networks. The Traditional deep learning models suffered from the vanishing gradient problem. It means when the depth of the network gets increased, gradients vanished while backward propagation. By incorporating the idea of skip connections into the ResNet architecture, the problem is resolved. It enhanced performance and made it easier for the gradients to propagate.

Panigrahi (2024) demonstrated that while deeper models have achieved higher accuracy on CIFAR-10 dataset, but the memory requirements and training time is significantly increased. Therefore, ResNet50 model architecture is recommended for fine-tuning on large-scale classification tasks in terms of accuracy and efficiency (Wang, 2022).

In addition, comparisons with VGG16, DenseNet, and MobileNet architectures showed that ResNet models beat (or achieve comparable performance with) other architectures, and they have better feature extraction and convergence speed than those other architectures (Simonyan, 2014). However, Howard (2017) highlighted that such comparisons often overlook inference speed and energy efficiency factors essential for real-world deployment. Table 1 shows the comparative summary of related work on deep learning models.

Benchmark datasets like CIFAR-10 and Fashion-MNIST both are widely used for image classification research. The CIFAR-10 dataset is developed by the Canadian Institute for Advanced Research (Huang, 2017). It has 60,000 color photos in ten categories which includes animals and cars (Krizhevsky, 2009). According to Xiao (2017), Fashion-MNIST, which is developed by Zalando Research, is a good substitute for the conventional MNIST dataset because it includes greyscale photos of clothing.

Moreover, class imbalance remains a significant challenge in supervised machine learning. The models that are trained on imbalanced data are more influenced towards the majority class (Johnson, 2019). To minimize the impacts of class imbalance, several strategies have been proposed such as dropout, regularization, weighted loss functions, data augmentation, class-weighted loss function, targeted loss, and SMOTE (Lin, 2017). Additionally, studies show that selection of the best-performing model depends heavily on computing efficiency. This also demonstrated that ResNet50 has high accuracy, studies comparing inference time, memory consumption, and energy efficiency indicate that optimization strategies are necessary to maintain efficiency in contexts with limited resources (Howard, 2017). Alam, (2024) shows that applying SMOTE with ResNet50 improved recall in diabetic retinopathy detection by 5.4%. Similarly, Ojo (2023) point out that when employing ResNet to detect plant diseases, GAN-based augmentation outperforms SMOTE.

Additionally, newer research suggests that simple tuning methods such as adjusting batch size, applying label smoothing, and optimizer selection can offer comparable gains to complex imbalance-handling methods (Shwartz-Ziv, 2023). This highlights the importance of strong baselines before introducing advanced solutions.

Moreover, recent studies have shown advancements in transfer learning with ResNet model. The pre-trained ResNet models fine-tuned on domain specific tasks has shown remarkable performance in medical imaging, image classification, remote sensing (Tan, 2019) . It shows that using pre-trained ResNet model can generalize better and reduce training cost which makes them more accessible for diverse use cases.

While some studies have investigated deep learning models for image classification, and have tackled the class imbalance problem using approaches such as SMOTE, focal loss and data augmentation, few of those studies have considered a simultaneous handling of the balanced and imbalanced situations. There are no end-to-end methods to assess how well a single model performs in both cases with common benchmarks here. In addition, prior work seldom focuses on the computational efficiency and scalability of these models as the class distributions are changed. In order to fill the gap, our study suggests a ResNet50-based architecture that can perform adequately in both balanced and imbalanced scenarios. Applying imbalance-handling techniques and model assessment on CIFAR-10 and Fashion-MNIST, we aim to provide a solution closer to reality and more generalizable for real-world classification tasks.

Proposed Framework

The proposed methodology is illustrated in Figure 1 for analyzing the performance of CIFAR-10 and Fashion-MNIST datasets on ResNet50. The workflow consists of dataset preprocessing, data balancing strategies, model training with various hyperparameters and performance evaluation.

Dataset Preparation and Preprocessing

CIFAR-10 comprises 60,000 RGB-size-32×32-images in 10 classes, and Fashion-MNIST contains 70,000 grayscale-28×28-size-images in 10 classes. Fashion-MNIST images are resized to 32×32 and converted to three channels by channel duplication, where the input images are gray in a single channel. This pre-processing ensures all inputs have consistent dimensions and it helps us to do a transfer learning using pre-trained ImageNet weights.

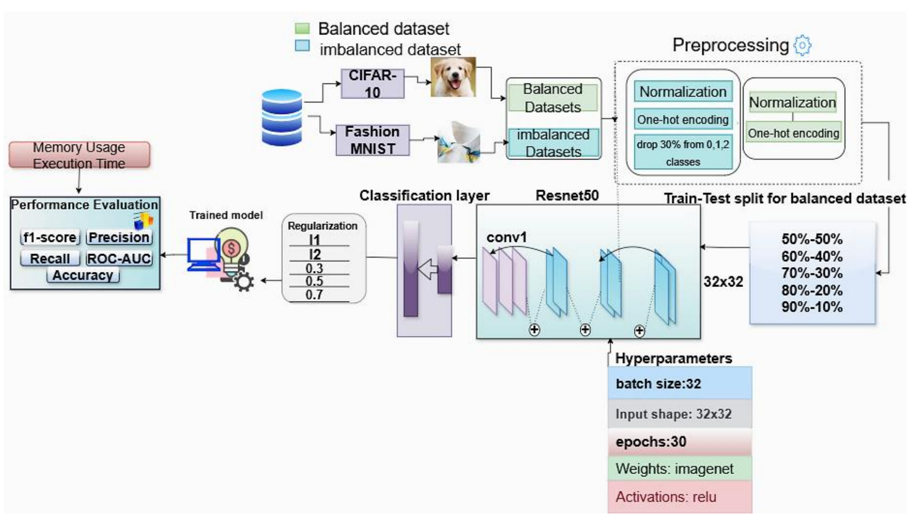

Figure 1: Illustration of the proposed methodology.

Normalization

In normalization, pixel values are normalized and scaled to [0,1] by dividing it by 255 to prevent the large pixel values from dominating the smaller ones.

Data Augmentation

To improve the model generalization and data diversity, data augmentation is applied to the balanced datasets. The augmentation techniques such as rotation and horizontal shift are applied to artificially expand the data. The width shift and height shift of 0.2 are applied to the dataset. Although dataset size is relatively large but its content is fixed and limited to specific lightening conditions.

No data augmentation is applied to imbalanced datasets, to maintain the synthetic class distribution and to isolate the effect of imbalance on performance without artificially boosting specific classes.

Class Imbalance

To simulate class imbalances, 10% of training samples are removed from classes 0, 1, and 2, resulting in a total reduction of 30%. This controlled, mild imbalance avoids severely distorting the dataset, while allowing us to evaluate model sensitivity to uneven classes. Although this method enables reproducible basic comparisons and helps to separate the effects of imbalance. More complex patterns such as long -tailed distributions are planned for future work. Class distribution is shown in tables 5 and 6 before and after imbalance.

Data Splitting

After preprocessing datasets are divided into train test splits with the ratio of 50%-50%, 60%-40%, 70%-30%, 80%-20%, and 90%-10%. These variations allow us to analyze the effect of increasing and decreasing the training and testing data on the model performance.

ResNet50

ResNet50 with modifications to the final layer to match the number of classes is used for classification. The input shape is set to 32x32 to align with the dataset dimension and the batch size is set to 32. Different regularization (l1, l2) and the dropout rate (0.1, 0.3, 0.5, 0.7) are applied. The performance of the model is evaluated using metrics such as F1-score, Precision, Recall, ROC-AUC, Accuracy, as well as memory usage and execution time.

Results and Discussion

Data Preparation and Model Configuration

The experiments are conducted on a PC with Intel Core i5-4210U CPU running at 1.70 GHz (boosting up to 2.40 GHz), 4 GB of installed RAM, 8 GB SSD, and a 64-bit operating system based on an x64 processor. The machine runs Windows 10 Pro, Version 22H2, with OS Build 19045. 5247.Google. Colab’s free tier is used with the Python version of 3.9.7, T4 GPU, 12GB RAM. The experiment utilizes libraries such as Tensorflow, Numpy, Scikit-Learn and Matplotlib. The datasets are available on TensorFlow/Keras’s built-in repositories.

CIFAR-10 is a state-of-the-art dataset for image classification, especially used to evaluate the performance of deep learning architectures. It is developed by the Canadian Institute for Advanced Research. It contains 50,000 training image samples of animals and 10,000 test samples and also has 10 classes (airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck). Each image is of 32x32 pixels in RGB format, having three color channels. The dataset has labels (0-9) corresponding to classes.

The Fashion-MNIST dataset is from the family of the original MNIST dataset. It contains 60,000 training images and 10,000 test images as shown in Table 1. The dataset has 10 classes, ranging from 0-9, containing images of clothing and accessories. The image size is 28x28 in the dataset. The images have an intensity value between 0 and 255.

The label categories in the dataset are T-shirts, trousers, pullovers, dresses, coats, sandals, shirts, sneakers, bags and ankle boots. Both are initially balanced but the class imbalance is introduced by removing 10% samples from 0, 1 and 2 classes, resulting in 30% drop. This resulted in 54,000 and 45,000 samples for CIFAR-10 and Fashion-MNIST respectively, as reflected in Table 2.

Table 1: Balanced Dataset (Before Drop).

Dataset	Total Samples	Classes (0-9) Samples per class
CIFAR-10	60,000	6,000
Fashion-MNIST	50,000	5,000

Table 2: Imbalanced Dataset (After Drop).

Dataset	Total Samples	Reduced classes (0-2) Samples per class	Unchanged Classes (3-9)
CIFAR-10	54,000	4,950	6,000
Fashion-MNIST	45,000	4,500	5,000

Different hyperparameter settings are applied to each dataset. CIFAR-10 used the ResNet50 model with pre-trained ImageNet weights, input shape (32, 32, 3), Adam optimizer, batch size of 32, and varied epochs. Data augmentation included random rotations, shifts, and horizontal flips. The list of CIFAR-10 hyperparameters for balanced and imbalanced datasets are shown in Tables 3 and 4.

Fashion-MNIST inputs are resized to (32, 32, 3) for compatibility, and similar augmentation techniques are applied. For imbalanced settings. A class reduction factor is introduced to evaluate the model's performance on imbalanced datasets, as shown in Tables 5 and 6.

Table 3: Hyperparameters for CIFAR-10 balanced Data.

Hyperparameter	Description	Value
Input shape	Defines the input size of images (32x32 with 3 color channels).	(32, 32, 3)
Weights	Uses pre-trained ImageNet weights to initialize model layers.	'imagenet'
Optimizer	Adam optimizer for efficient training by adjusting learning rate.	Adam()
Batch size	Number of samples processed together in each model update.	32
Rotation range	Rotates images randomly within the specified angle for augmentation.	20
Width shift range	Randomly shifts images horizontally for data augmentation.	0.2
Height shift range	Randomly shifts images vertically to augment the dataset.	0.2
Activation	RELU activation for hidden layers, softmax for final classification.	relu,'softmax
Epochs	Number of iterations	25

Table 4: Hyperparameters for CIFAR-10 imbalanced Data.

Hyperparameter	Description	Value
Imbalance factor	Defines the extent of class imbalance in the dataset.	0.1
Imbalanced classes	Classes to reduce in the dataset for imbalance.	[0, 1, 2]
Epochs	Number of iterations to train the model	25
Number to remove	Number of samples to remove for class imbalance.	5000
Base model	ResNet50 model with pre-trained ImageNet weights.	weights='imagenet',
Predictions	Output layer with softmax activation for classification.	Dense(10, activation='softmax')
Datagen	Image data augmentation for improving model generalization.	rotation_range=20

Table 5: Hyperparameters for Fashion-MNIST balanced Data.

Hyperparameter	Description	Value
Epochs	Number of training epochs for the model	5
Batch Size	Number of samples per gradient update	32
Optimizer	Optimization algorithm used to minimize loss function	Adam
Loss Function	Loss function used for training	Categorical Crossentropy
Metrics	Performance metric to evaluate during training	Accuracy
Rotation Range	Degree range for random rotations of images	20
Width Shift Range	Fraction of the total width for random horizontal shift	0.2
HeightShift Range	Fraction of the total height for random vertical shift	0.2
Horizontal Flip	Randomly flip images horizontally	True
Input Shape	Input shape of the images fed to the model	(32, 32, 3)

Table 6: Hyperparameters for Fashion-MNIST imbalanced Data.

Hyperparameter	Description	Value
Epochs	Number of training epochs for the model	7
Batch Size	Number of samples per gradient update	32
Optimizer	Optimization algorithm used to minimize loss function	Adam
Loss Function	Loss function used for training	Categorical Crossentropy
Metrics	Performance metric to evaluate during training	Accuracy
Input Image Size	Resized image size for the model input	32x32
Imbalance Factor	Factor for simulating imbalance in the dataset	0.5
Base Model	Pre-trained model used for transfer learning	ResNet50
Freeze Base Model Layers	Whether to freeze the layers of the pre-trained model	True
Dense Layer	Number of neurons in the fully connected layer	1024

Evaluation Metrics and Performance Comparison

The model is evaluated using Accuracy, Precision, Recall, F1-score, and ROC Curve (AUC) as shown in Table 7. The performance of the model is improved in the imbalanced setting, which is initially counter intuitive. For CIFAR-10, the F1-score increased from 0.36 to 0.72, and AUC improved from 0.65 to 0.68. While for Fashion-MNIST the F1-score increased from 0.59 to 0.62 and AUC from 0.84 to 0.86.

Table 7: Performance of Fashion-MNIST and CIFAR-10 datasets.

Dataset	Metric	Balanced	Imbalanced
CIFAR-10	F1 Score	0.36	0.72
	Precision	0.40	0.62
	Recall	0.37	0.66
	Accuracy	0.67	0.73
	ROC	0.65	0.68
Fashion-MNIST	F1 Score	0.59	0.62
	Precision	0.63	0.65
	Recall	0.73	0.77
	Accuracy	0.66	0.86
	ROC	0.84	0.86

These results can be interpreted in several ways. First, moderate imbalance may push the model to focus more on the minority classes, especially when augmented data helps preserve their variance. Second, since some classes are semantically or visually similar, the model may benefit from having fewer confusing examples in classes with reduced samples. The CIFAR-10 matrix reveals various confusions among semantically related categories such as Dog, Frog, and Cat, which is a common issue in object recognition tasks is shown in Figure 3. These matrices lend partial support to our hypothesis that a slight imbalance may improve generalization by decreasing the frequency of overlapping instances within particular classes.

Even with a 10% decrease in data per class, the model shows reasonably strong predictions for the reduced classes (T-shirt/top, Trouser, Pullover) in the Fashion-MNIST matrix (see Figure 4). This suggests that the model's ability to identify distinctive characteristics for these under-represented classes has not diminished. However, there are still misclassifications between visually related categories, including shirt and T-shirt/top or sneaker and ankle boot, indicating that model performance is still impacted by visual similarities.

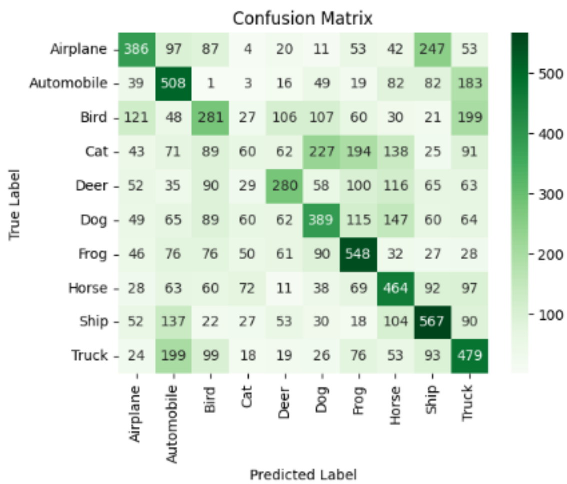

Figure 2: Confusion Matric (CIFAR-10 dataset).

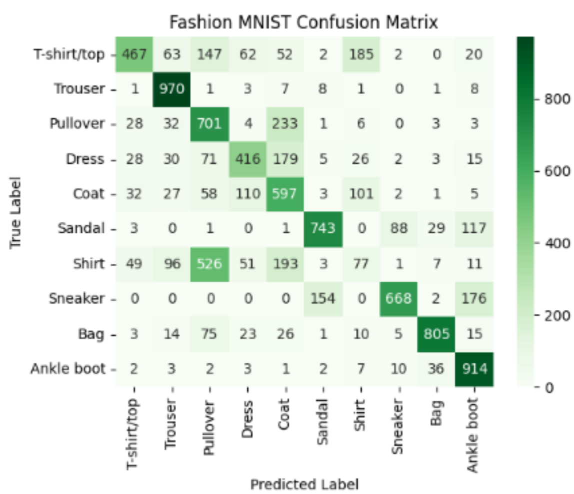

Figure 3: Confusion Matrix (Fashion-MNIST dataset).

Computational Resource Usage

Several significant distinctions between the datasets and class balance schemes when training and testing times and memory use are analyzed as shown in Table 8. Due to the complexity that imbalanced data introduces, CIFAR-10 takes much longer to train in the imbalanced setting (2163.77 seconds) than in the balanced one (1496.23 seconds). Similar trends are shown in Fashion-MNIST; however, training durations are somewhat shorter (355.95 seconds balanced vs. 471.38 seconds imbalanced). For both datasets, memory utilization increases significantly in the unbalanced settings as well; CIFAR-10 uses 5242.31 MB, whereas Fashion-MNIST uses 9627.61 MB. These results highlight the resource requirements and computational cost involved in managing imbalanced datasets, highlighting the necessity of good model designs to properly handle these difficulties.

Table 8: Training/Testing Time, Time Complexity and memory usage on CIFAR-10 and Fashion-MNIST datasets.

Dataset	Metric	Balanced	Imbalanced
CIFAR-10	Training Time	1496.23 sec	2163.77 sec
	Testing Time	3.5 sec	3.5 sec
	Time Complexity (Training)	O(n * m * p)	O(n * m * p)
	Time Complexity (Testing)	O(k * p)	O(k * p)
	Memory Usage	2553.05 MB	5242.31 MB
Fashion-MNIST	Training Time	355.95 sec	471.38 sec
	Testing Time	134.9 sec	134.9 sec
	Time Complexity (Training)	O(n * m * p)	O(n * m * p)
	Time Complexity (Testing)	O(k * p)	O(k * p)
	Memory Usage	4868.14 MB	9627.61 MB

Impact of Train/Test Splits

To evaluate model generalization under different scenarios, we applied multiple train/test splits. The results of different training and testing splits on model performance can be seen the Table 9. It can be seen that as the percentage of training grows, test accuracy for CIFAR-10 often increases. The performance of model is better on Fashion-MNIST, comparatively better than CIFAR-10 in all split settings. It is clear in the Table 9 that the CIFAR-10 achieved 0.36% accuracy with 70/30 split and the other achieved 0.67% with 60/40 split. The results also suggest that small training sets can significantly hurt performance but slightly larger test sets such as 40% still yield reliable evaluations.

It is proven from the previous results that the model performs better on Fashion-MNIST compared to CIFAR-10. The Fashion-MNIST dataset is further evaluated with different Regularization techniques such as l1, l2 and dropout with rates of 0.3,0.5 and 0.7 as the results can be seen in Table 10. For Fashion-MNIST's limited diversity and very shallow depth, these methods are chosen to mitigate the risk of overfitting. L1 and L2 regularization helps to penalize high weights and encourage generalization, whereas dropout randomly disables neurons during training to prevent co-adaptation and improve robustness.

Table 9: Results of different splits settings.

Dataset	Train/Test Split	Test Accuracy	Training Time (s)	Testing Time (s)	Memory Usage (MB)
CIFAR-10	50% / 50%	0.3113	133.61	127.95	3228.08
	60% / 40%	0.3377	129.26	122.01	2347.43
	70% / 30%	0.3625	98.56	3.50	2429.31
	80% / 20%	0.7376	130.05	3.2	2347.43
	90% / 10%	0.8577	98.56	3.50	2347.43
Fashion-MNIST	50% / 50%	0.6358	259.38	134.9	2927.45
	60% / 40%	0.6725	252.63	5.73	2428.73
	70% / 30%	0.6462	292.54	4.00	2543.84
	80% / 20%	0.7313	344.11	5.45	2401.57
	90% / 10%	0.8607	263.73	1.38	2541.29

Table 10: Results of Regularization Techniques for Fashion-MNIST.

Regularization	Accuracy	AUC	Training Time (s)	Testing Time (s)	Memory Usage (MB)
L1 Regularization	0.3860	0.8803	375.18	2.84	2652.21
L2 Regularization	0.6286	0.9348	372.81	2.85	3720.61
Dropout (0.3)	0.6606	0.9447	490.34	2.54	4499.43
Dropout (0.5)	0.6129	0.9388	398.42	2.72	4725.57
Dropout (0.7)	0.5627	0.9207	373.61	2.87	5128.71

The varying impacts of regularization techniques can be seen in Figure 4. L1 regularization achieved an AUC of 0.8803 and an adequate test accuracy of 0.3860. A significant enhancement is observed with an AUC of 0.9348 and a test accuracy of 0.6286 after L2 regularization. Furthermore, dropout rates affected model performance; the maximum test accuracy of 0.6606 and AUC of 0.9447 are attained with a dropout rate of 0.3. The results demonstrate that a dropout rate of 0.3 achieves the ideal compromise between accuracy and generalization, but dropout with a rate of 0.5 and L2 regularization also shows strong performance.

Table 11 presents a comparative overview of the accuracies obtained by similar deep learning techniques in comparison to our suggested method. He et al. (2016) achieved 76% accuracy on CIFAR-10 via ResNet with a good baseline but not using class imbalance handling. The resnet-18 *is* used by the Panigrahi (2024) and it achieved an accuracy of 70%. The performance is slightly low because of shallow architecture. Howard et al. (2017) reported, 68% for MobileNet. Alam et al. (2024) achieved about 82% accuracy on imbalanced medical images using ResNet-50 combined with SMOTE, but their results are domain-specific and not generalize to other datasets.

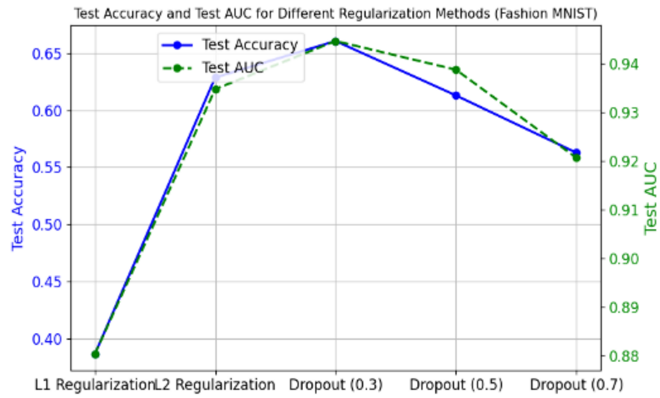

Figure 4 : Impact of L1, L2 and Dropout (0.3,0.5 and 0.7) on Fashion-MNIST Dataset.

Table 11: Comparison of the proposed model with the existing approaches.

Reference	Method	Accuracy %
He et al. (2016)	ResNet	76 (CIFAR-10)
Panigrahi (2024)	ResNet-18	70 (CIFAR-10)
Alam et al. (2024)	ResNet-50+SMOTE	82 (Medical images)
Howard et al. (2017)	MobileNet	68 (CIFAR-10)
Proposed Model	ResNet-50	73(CIFAR-10)/ 86 (Fashion-MNIST)

Comparatively, the proposed model achieved 73% accuracy on CIFAR-10 and 86% on fashion-MNIST, showing competitive performances. Interestingly, our method specifically explores the effect of class imbalance, which is not mainly addressed in baseline methods. This reflects the strength of our approach in maintaining stable accuracy while dealing with class-imbalance, thus providing better utility for real -world applications where data distribution is rarely uniform.

Conclusion

This study investigated the performance of ResNet50 on two benchmark datasets, CIFAR-10 and Fashion-MNIST, under both balanced and imbalanced conditions. Different metrics are used to evaluate the performance of model such Training/testing time, memory consumption, accuracy, precision, recall, F1-score, and time complexity. The result indicated that the model performed better in the Fashion-MNIST dataset than in the CIFAR-10 dataset in all cases. Remarkably, model worked better than raw data in imbalanced setting where CIFAR-10 F1-score increasing from 0.36 to 0.39, Fashion-MNIST from 0.59 to 0.62. These findings illustrated ResNet50's robustness in dealing with class imbalance, especially when employed with suitable preprocessing, and hyperparameter tuning. The superior results are obtained over imbalanced data sets at the cost of a higher computational burden notably in terms of memory overhead and training time. For example, training times in CIFAR-10 increased from 1496.23 s (balanced) to 2163.77 s (imbalanced) with about twice the memory consumption. Fashion-MNIST showed similar trends, only with less overall training time. Further comparisons are made against model accuracy and computational efficiency with the help of various regularization techniques. These results may prove the necessity of balanced design in architecture as well as in dataset preparation for optimizing performance and resource utilization in real-world applications. This study is limited by the use of a fixed imbalance strategy and the absence of a validation set for hyperparameter tuning. Results are based on only two datasets, which may affect generalizability. Future work will explore more realistic imbalance scenarios, include validation sets, and extend the analysis to additional models and datasets.

References

Alam, M. Z. (2024). Enhancing Transfer Learning for Medical Image Classification with SMOTE: A Comparative Study. *arXiv preprint arXiv:2412.20235.*

Bello, I. F. (2021). Revisiting resnets: Improved training and scaling strategies. *Advances in Neural Information Processing Systems*, 34, 22614-22627.

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

He, K. Z. (2016). Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 770-778).

Hossain, M. B. (2022). Transfer learning with fine-tuned deep CNN ResNet50 model for classifying COVID-19 from chest X-ray images. *Informatics in Medicine Unlocked*, 30, 100916.

Howard, A. G. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. . *arXiv preprint arXiv:*, 1704.04861.

Huang, G. L. (2017). Densely connected convolutional networks. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. (pp. 4700-4708)).

Johnson, J. M. (2019). Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1), 1-54.

Krizhevsky, A. &. (2009). Learning multiple layers of features from tiny images.

Lin, T. Y. (2017). Focal loss for dense object detection. . *In Proceedings of the IEEE international conference on computer vision*, (pp. (pp. 2980-2988)).

Ojo, M. O. (2023). Improving Deep Learning Classifiers Performance via Preprocessing and Class Imbalance Approaches in a Plant Disease Detection Pipeline. *Agronomy* , 13(3), 887.

O'Mahony, N. C. (2020). Deep learning vs. traditional computer vision. In Advances in computer vision. *proceedings of the 2019 computer vision conference (CVC)* (pp. 128-144). Springer International Publishing.

Panigrahi, U. S. (2024). A ResNet-101 deep learning framework induced transfer learning strategy for moving object detection. . *Image and Vision Computing,* , 146, 105021.

Shwartz-Ziv, R. G. (2023). Simplifying neural network training under class imbalance. *Advances in Neural Information Processing Systems, 36,*, 35218-35245.

Simonyan, K. &. (2014). Very deep convolutional networks for large-scale image recognition. . *arXiv preprint arXiv*, :1409.1556.

Tan, M. &. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. *In International conference on machine learning* (pp. (pp. 6105-6114)). PMLR.

Wang, G. L. (2022). A network combining a transformer and a convolutional neural network for remote sensing image change detection. *Remote Sensing,*, 14(9), 2228.

Xiao, H. R. (2017). Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:*, 1708.07747.