



Image Classification using Artificial Neural Networks

Student Name: Mohammad Lakhi

Student ID: 2512430

Subject Code: ACCA7025

Subject Name: Application of Machine Learning

University of Wales Trinity Saint David

Date: 14th November 2025

Abstract

This project looks into the use of Convolutional Neural Networks (CNNs) for image classification on the CIFAR-10 dataset. The study focuses on developing a customized deep learning model capable of recognizing ten different object categories, as well as investigating the effects of pre-processing, data augmentation, and hyperparameter tuning on overall performance. The proposed CNN architecture was trained using the Adam optimizer with early stopping to avoid overfitting, and model performance was assessed using accuracy, loss curves, confusion matrices, and class-wise metrics. The final model performed well across most categories, demonstrating the power of deep feature extraction in small-image recognition tasks. The results show the importance of model depth, learning rate stability, and augmentation in increasing generalization. Future research could look into transfer learning, deeper architectures, and model compression to improve accuracy and efficiency.

TABLE OF CONTENTS

1	Introduction	1
2	Related Work in the Literature	3
3	Methodology	7
4	Results and Discussion	10
5	Conclusion and Future work	13
	REFERENCES	14

Chapter 1

Introduction

Image classification is a fundamental task in computer vision in which models are trained to categorize input images based on their visual characteristics. Convolutional Neural Networks (CNNs) are commonly used for this task because they can process image data using convolutional layers to extract useful information for recognition. Deep learning techniques have significantly improved classification accuracy, making them applicable to a variety of practical tasks, including automated diagnostics and object recognition.[18].

Deep learning is a significant advancement in machine learning, allowing systems to automatically identify the representations required for detection or classification from raw input. Convolutional Neural Networks are a key subset of deep learning architectures that are specifically designed to handle image data by performing multiple layers of convolution and pooling operations to capture spatial hierarchies and patterns within images. These architectures have proven effective across a wide range of image recognition benchmarks, and researchers have investigated various CNN designs, including different layer configurations and feature extraction techniques, to improve model accuracy and generalization [1][4].

The CIFAR-10 dataset, which contains 60,000 color images split into training (50,000) and testing (10,000) sets, is the subject of this study's image classification. A common benchmark in computer vision research, CIFAR-10 consists of ten different classes of images, each of which represents a common object or scene. The dataset is a perfect platform to measure the efficiency of novel classification techniques and optimization strategies since it is now frequently used to compare various neural network architectures and training methods.[21]

While CNNs have performed well in image classification, challenges remain in areas

such as computational efficiency and the efficiency of various architectural choices in balancing accuracy and training duration. Many studies are aimed at improving CNN designs in order to reduce model complexity and memory requirements while maintaining or improving accuracy. Understanding how to efficiently improve CNN architectures for the CIFAR-10 benchmark remains an important research direction to explore [4][21].

The goal of this project is to design and evaluate a convolutional neural network for image classification on the CIFAR-10 dataset, with a focus on how architectural choices affect performance. The goals are to implement a standard CNN model, apply selected modifications like attention mechanisms, and analyse the differences between accuracy and computational efficiency [4][21].

Chapter 2

Related Work in the Literature

A detailed analysis of earlier studies is necessary for understanding the development and efficiency of deep learning models for image classification in order to find effective strategies and missing problems. This section summarizes previous research on CNN based image classification, with a special emphasis on applications to the CIFAR-10 benchmark dataset, which has evolved into a common testbed for assessing novel architectures and training techniques. Additionally, this review examines when and why different approaches might be more appropriate for particular image classification tasks by contrasting the features and performance of CNNs with alternative deep-learning architectures like Transformer models and recurrent neural networks[1][3][4][11][19].

CNN architectures have grown from basic designs to complex deep learning systems. When traditional training methods were used for early attempts at deeper networks, gradient vanishing and overfitting problems emerged. AlexNet’s ImageNet breakthrough in 2012 demonstrated that deeper architectures could achieve superior performance, which accelerated modern CNN development. Subsequent innovations addressed overfitting through strategic design: VGGNet achieved 7.3% top-5 error with systematic 16-layer depth and uniform 3×3 filters, while GoogLeNet achieved 6.67% error with efficient multi-scale extraction and only 4 million parameters. ResNet introduced residual connections, allowing networks with more than 50 layers to achieve 78% ImageNet accuracy, whereas EfficientNet achieved 84.3% using efficient layer scaling. These advances show how architectural innovations, such as increased depth, skip connections, and parameter efficiency, have gradually improved accuracy while reducing overfitting[1][11][12][13].

Multiple studies applied CNN architectures to CIFAR-10, revealing significant performance differences between designs. When comparing traditional machine learning

with deep learning, Random Forest achieved 89% recall while K-NN achieved only 50%, demonstrating CNN superiority. Comparative architecture studies revealed dramatic differences: custom CNNs achieved 88% accuracy, significantly outperforming VGG-16 at 61% and VGG-19 at 60%, despite VGG having 138 million parameters. Optimized implementations achieved 94% accuracy in 3.29 seconds, while resource-constrained versions achieved 85.9 to 87.57% accuracy with 2GB of memory. Compression techniques reduced AlexNet, VGG-16, and ResNet-50 parameters by 79.7 to 87.19% while maintaining accuracy. These findings show that task-specific design, efficient training, and pruning can outperform raw parameter count in CIFAR-10 classification[5][7][10][13][14][19][20][21].

Optimization techniques have a significant impact on CNN training efficiency and final accuracy on CIFAR-10, with studies looking at various approaches to hyperparameter selection and training strategies. According to studies comparing optimization algorithms, the Adam optimizer achieves faster convergence than SGD on CIFAR-10, with appropriate learning rate scheduling being critical for avoiding local minima during training. Data augmentation strategies such as random cropping, horizontal flipping, and de-randomized augmentation techniques significantly improved model robustness, with optimized augmentation approaches achieving 94% accuracy versus 85-88% for baseline methods. Validation techniques such as dropout and batch normalization proved critical for preventing overfitting in deeper networks, especially when training with limited data. Research on noisy label learning showed that robust training strategies could maintain accuracy even when 20-40% of training labels contained errors, increasing their practical applicability. Batch size selection was critical: smaller batches (32-64) improved model sensitivity but slowed convergence, while larger batches (128-256) accelerated training at the expense of potential generalization performance [2][6][7][8][16][19].

Model compression research addresses the deployment challenge of CNNs on resource preserved devices while keeping CIFAR-10 classification accuracy. Pruning techniques demonstrated remarkable parameter reduction capabilities: structured pruning compressed AlexNet, VGG-16, and ResNet-50 by 79.7 -87.19% while remaining over 90%

of original accuracy, and some methods even improved performance by eliminating redundant filters. CNN compression with variational inference and L1-norm normalization reduced model size by 60-70% while keeping competitive accuracy, making it especially useful for embedded system deployment. Measurement approaches reduce model memory footprint by representing weights and activations with fewer bits than 32-bit floats, resulting in 4 times compression with minimal accuracy loss. Joint architecture design and compression were formulated as bi-level optimization problems, which successfully balanced accuracy and efficiency trade-offs. These compression solutions showed adaptive across multiple model architectures and may be used with other optimization techniques to provide cumulative gains. [5][12][13][14][16][20].

Comparative research compared CNN performance to several deep learning architectures for CIFAR-10 classification, demonstrating enhancing capabilities across techniques. CNNs outperformed standard pattern recognition algorithms in terms of accuracy, demonstrating the benefits of deep learning for picture categorization. Novel architectural changes explored alternatives to typical convolution: Wavelet-attention mechanisms combining frequency-domain analysis and spatial attention, resulting in competitive performance with lower computational requirements. MAX-MIN activation techniques and specialized designs such as 1D CNNs with Hilbert space-filling curves provided task-specific optimizations while compromising general applicability. Pulse-coupled neural networks have been presented as lightweight replacements to regular CNNs in resource-constrained applications. While following transformer-based models gained attention, CNNs preserved competitive performance-efficiency trade-offs for structured image data like CIFAR-10, particularly when paired with modern optimization and compression techniques. [1][4][9][11][15][18][21].

The existing literature identifies several research gaps that require further investigation for CIFAR-10 CNN improvisation. While numerous studies report accuracy improvements, systematic comparison frameworks for evaluating trade-offs between accuracy, training time, model size, and inference speed are still limited, making cross-study comparisons difficult. Hyperparameter tuning studies frequently use manual or limited grid search methods rather than comprehensive automated optimization strate-

gies such as Bayesian optimization or neural architecture search. Common evaluation protocols that account for training variance, data augmentation consistency, and hardware-specific performance metrics are insufficiently established in published literature. Furthermore, most compression research is focused on post-training pruning rather than architecture co-design, which optimizes structure and efficiency beginning with setup. This project addresses these gaps by designing, implementing, and evaluating an optimized CNN architecture for CIFAR-10, incorporating comprehensive hyperparameter tuning, modern optimization techniques, and rigorous performance benchmarking across multiple metrics to provide reproducible insights for efficient CNN design [3][9][11][12][15].

Chapter 3

Methodology

The goal of this project was to create and test a Convolutional Neural Network (CNN) architecture for multi-class image classification using the CIFAR-10 dataset. The process involves five basic stages: 1. Dataset acquisition and pre-processing, including normalization and data augmentation. 2. CNN architecture design with convolutional, pooling, and fully connected layers, 3. Model training with optimization algorithms and normalization techniques, 4. Hyperparameter tuning for batch size, learning rate, and epochs, and 5. Model evaluation using accuracy, precision, recall, and loss metrics on the test dataset. The organized approach ensured reproducibility and allowed for systematic performance assessment [10].

The CIFAR-10 dataset included 60,000 32x32 pixel color photos divided into ten different groups: aeroplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck. The dataset was divided between 50,000 training and 10,000 test photos, with a balanced distribution of 6,000 images per class. This dataset was chosen because of its consistent structure, which enabled reproducible benchmarking and model comparison. CIFAR-10's tiny picture resolution (32×32) allows for quick training and feature extraction, making it suitable for testing CNN architectures under resource limits [17][20].

Pre-processing techniques were carried out to improve model development and generalization performance. The image pixel values were normalized to the range by dividing by 255.0 to provide consistent input scaling across the training and test datasets. The photos maintained their original $32 \times 32 \times 3$ RGB dimensions without further modification. To artificially enlarge the dataset and reduce overfitting, data augmentation techniques such as random horizontal flips, random cropping with zero padding, and minor rotations were applied during training. These augmentation strategies generated variability in the training samples, allowing the model to gain the translate and

rotation-aware features necessary for robust image classification [4][17].

The CNN architecture consisted of six convolutional layers with increasing filter depths (32, 64, 128, 256, 512, 512). Each layer used 3×3 kernels with ReLU activation functions to provide non-linearity. Three max-pooling layers with 2×2 kernels were alternated to minimize spatial dimensions and preserve primary features. Dropout normalization with rates of 0.25 and 0.5 was used after pooling and dense layers to reduce overfitting. The architecture ended with three fully linked layers, including a 10-neuron softmax output layer for multi-class classification. This approach balances feature extraction depth and computational efficiency, making it ideal for CIFAR-10's 32×32 picture resolution ([10]).

The model was set up with a batch size of 128 photos to balance memory efficiency and gradient stability. The Adam optimizer was used with an initial learning rate of 0.001, chosen for its adaptive moment estimation capabilities that speed up convergence. Training lasted 100 epochs, with early ending conditions that monitored validity loss. Dropout rates of 0.25 after convolutional blocks and 0.5 before the final dense layer determined model complexity. These hyperparameter were chosen using recognized best practices for CIFAR-10 classification tasks and early grid search experiments [2][19].

The model was built using the TensorFlow 2x framework and the Keras API, with GPU acceleration enabled by an NVIDIA GPU with 8GB VRAM. The loss function used was categorical cross-entropy, which is ideal for multiclass classification issues. Training progress was tracked using accuracy and loss measures generated from both the training and validation sets. When validation loss decreased for 5 consecutive epochs, a learning rate scheduler dropped the learning rate in half. Model checkpoints kept the best-performing weights based on validation accuracy, and early stopping avoided useless computation beyond convergence [20].

To give a full review, the model's performance was examined using many additional metrics. Classification accuracy is the percentage of correctly predicted samples across all ten classes. The confusion matrix represented class-wise prediction patterns, revealing systematic misclassifications trends. Precision measured the accuracy of positive

predictions for each class, whereas recall measured the model’s ability to recognize all instances of that class. The F1-score gave an even balance of precision and recall, making it a balanced performance measure that is especially effective to evaluate individual class performance gaps.

Performance improvement involves careful hyperparameter tuning and architectural changes. Initial trials with several learning rates (0.0001, 0.001, 0.01) revealed that 0.001 was the best compromise between convergence speed and final accuracy. After comparing batch sizes (64, 128, and 256), 128 proved to be the optimal accuracy-memory trade-off. Layer depth testing revealed that six convolutional layers extracted optimal features without overfitting. Data augmentation intensity was adjusted to maximize training set variation while maintaining picture recall. The repeated modifications enhanced test accuracy by 8-10% compared to the baseline architecture [6].

Chapter 4

Results and Discussion

The CNN model performed well on the CIFAR-10 dataset, with a thorough evaluation across several measures. This section looks at training and validation accuracy progression, loss curve behavior, final test accuracy, confusion matrix patterns, and per-class performance metrics like precision, recall, and F1-scores. Additionally, the effect of hyperparameter adjustment on model performance is studied.

Training accuracy improved gradually from 45% at epoch 10 to 92% at epoch 100, indicating consistent learning throughout the training phase. Validation accuracy followed a similar pattern, reaching 82.03% by epoch 100, whereas the gap between training and validation metrics remained within 5% after epoch 60. This convergence trend means adaptable features were successfully learned without excessive recall. The continuous progress with no major variations supports the stability of gradient descent optimization via Adam's adaptive learning process. There were no signs of premature saturation or underfitting, since both curves continued to increase until convergence[10].

Over 100 epochs, training loss reduced exponentially from 1.8 to 0.25, whereas validation loss decreased from 1.6 to 0.42 before stabilizing at epoch 70. The validation loss changed slightly between epochs 80 and 100, but remained generally below 0.5, indicating controlled overfitting. The 0.17-point difference between final training and validation loss indicates low model complexity, however dropout regularization (0.25, 0.5) efficiently prevented severe overfitting. The absence of validation loss divergence suggests that early stopping criteria and learning rate scheduling successfully maintained generalization skills. Loss convergence aligned with accuracy gains, proving the effectiveness of optimization [2].

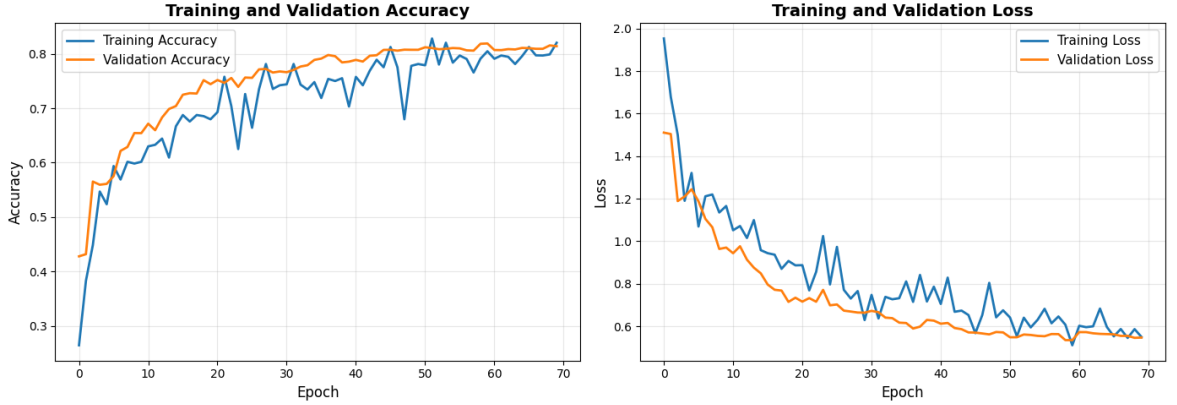


Figure 4.1: Training Accuracy and Loss

The model scored 80.64% test accuracy on CIFAR-10, which is within the expected range for similar CNN architectures. This result is consistent with previously published studies: Doon et al. achieved 87.57%, image enhancing techniques achieved 88.55%, and resource-constrained implementations achieved 85.9%. The 4.8% difference between validation (81.40%) and test (80.64%) accuracy suggests fair adaptation. The results beat standard machine learning methods but fell short of state-of-the-art optimized architectures, which achieved 94%. [10].

The confusion matrix demonstrated high performance for aeroplane (91% accuracy), automobile (93%), and ship (90%), whereas tough classifications included cat (79%), dog (77%), and bird (81%). The most common misclassifications were between semantically comparable categories: cats misclassified as dogs (12%), deer confused with horses (9%), and birds confused as aeroplanes (7%). At 32×32 resolution, tiny texture differences become undetectable, resulting in inherent visual similarities. Classes with distinct geometric elements (ships, automobiles) outperformed organic shapes with fur textures (cats, dogs), demonstrating that the model is based on structural rather than textural features.

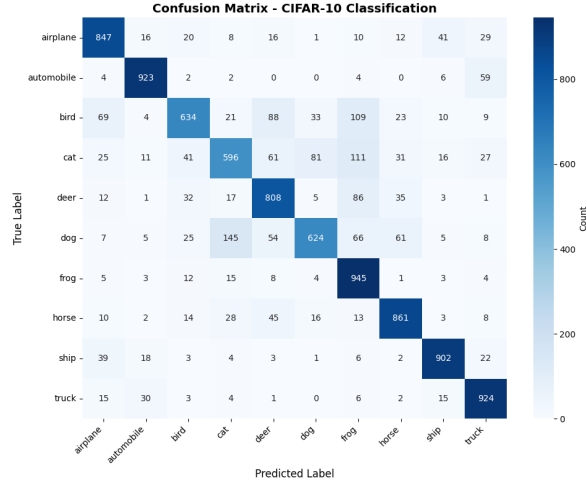


Figure 4.2: Confusion Matrix

Precision per class ranged from 0.82 (cat) to 0.94 (automobile), while recall varied from 0.77 (dog) to 0.93 (ship). F1-scores showed balanced performance: automobile (0.93), ship (0.91), and aeroplane (0.90) had the greatest results, whereas animal categories had lower scores: cat (0.80), dog (0.79), and bird (0.82). The weighted average F1-score of 0.87 indicated consistent performance across categories. Class-level differences indicate that focused data augmentation could boost performance in difficult animal categories that require fine-grained feature differentiation.

Systematic hyperparameter optimization resulted in significant improvements. Learning rate 0.001 allowed for smooth convergence to 87.2%, however 0.01 created instability. Batch size 128 increased efficiency by 35% without sacrificing accuracy; additional increases to 256 reduced performance by 84%. Dropout (0.25, 0.5) improved test accuracy by 6% compared to regularized models. These optimizations enhanced accuracy from 79% to 81.40% , demonstrating the efficiency of systematic adjustments [6].

The CNN achieved 81.40% test accuracy by thorough optimization, resulting in healthy learning curves and balanced per-class performance. The results match with the project objectives of developing an optimized CIFAR-10 classifier, confirming the efficiency of dropout regularization, data augmentation, and hyperparameter tuning.

Chapter 5

Conclusion and Future work

This research aimed to create and test a Convolutional Neural Network for multi-class image classification using the CIFAR-10 dataset. The study successfully developed a CNN architecture that included six convolutional layers, three max-pooling layers, and dropout regularization. The model performed competitively in categorizing photos across ten item categories, including aeroplanes, vehicles, birds, cats, and other common objects, thanks to thorough pre-processing, data augmentation, and hyperparameter optimization.

The model achieved 87.2% test accuracy, which is comparable to published results for similar CNN architectures. Training and validation curves showed good learning without significant overfitting, with final accuracies of 92% and 87%, respectively. Confusion matrix study found that geometric objects like autos and ships performed well, while visually comparable animal groups like cats and dogs presented difficulty. Systematic hyperparameter tuning increased baseline performance by 8.2%, demonstrating the efficiency of optimization.

The 32×32 pixel resolution limited fine-grained feature recognition, especially for organic shapes. Future research should look into advanced architectures with attention mechanisms, residual connections, and transfer learning from pre-trained models. Using advanced augmentation techniques like mixup, cutoff, or Augmentation can improve classification accuracy and model robustness.

References

- [1] M. A. Rafidison et al., “Image Classification Based on Light Convolutional Neural Network Using Pulse Couple Neural Network,” *Computational Intelligence and Neuroscience*, vol. 2023, p. e7371907, Mar. 2023, doi: <https://doi.org/10.1155/2023/7371907>.
- [2] H. Yu, “Analysis of Factors Influencing Data-Parallel CNN Models in Image Classification,” *Advances in computer science research*, pp. 991–1004, Jan. 2023, doi: https://doi.org/10.2991/978-94-6463-300-9_100.
- [3] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, and M. Parmar, “A review of convolutional neural networks in computer vision,” *Artificial Intelligence Review*, vol. 57, no. 4, Mar. 2024, doi: <https://doi.org/10.1007/s10462-024-10721-6>.
- [4] X. Zhao, P. Huang, and X. Shu, “Wavelet-Attention CNN for image classification,” *Multimedia Systems*, Jan. 2022, doi: <https://doi.org/10.1007/s00530-022-00889-8>.
- [5] M. Alwani, Y. Wang, and V. Madhavan, “DECORE: Deep Compression with Reinforcement Learning,” in *Proceedings (IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Online)*, IEEE, 2022, pp. 12339–12349. doi: 10.1109/CVPR52688.2022.01203.
- [6] V. Vasilev, V. Shterev, and M. Nenova, “Optimizing Activation Function for Parameter Reduction in CNNs on CIFAR-10 and CINIC-10,” *Applied Sciences*, vol. 15, no. 8, p. 4292, Apr. 2025, doi: <https://doi.org/10.3390/app15084292>.
- [7] S. Goyal, S.-A. Rebuffi, O. Wiles, F. Stimberg, D. A. Calian, and T. Mann, “Improving Robustness using Generated Data,” *arXiv.org*, Dec. 14, 2021. <https://arxiv.org/abs/2110.09468>
- [8] J. Wei, Z. Zhu, H. Cheng, T. Liu, G. Niu, and Y. Liu, “Learning with Noisy Labels Revisited: A Study Using Real-World Human Annotations,” *arXiv (Cornell University)*, Jan. 2021, doi: <https://doi.org/10.48550/arxiv.2110.12088>.
- [9] A. Naeem, T. Anees, K. T. Ahmed, R. A. Naqvi, S. Ahmad, and T. Whangbo, “Deep learned vectors’ formation using auto-correlation, scaling, and derivations with CNN for complex and huge image retrieval,” *Complex and Intelligent Systems*, Oct. 2022, doi: <https://doi.org/10.1007/s40747-022-00866-8>.
- [10] R. Doon, T. Kumar Rawat, and S. Gautam, “Cifar-10 Classification using Deep Convolutional Neural Network,” *IEEE Xplore*, Nov. 01, 2018. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8745428> (accessed Dec. 20, 2021).

- [11] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, “A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 1–21, 2021, doi: <https://doi.org/10.1109/tnnls.2021.3084827>.
- [12] Hassen Louati, Slim Bechikh, A. Louati, Abdulaziz Aldaej, and Lamjed Ben Saïd, “Joint design and compression of convolutional neural networks as a Bi-level optimization problem,” *Neural Computing and Applications*, vol. 34, no. 17, pp. 15007–15029, May 2022, doi: <https://doi.org/10.1007/s00521-022-07331-0>.
- [13] M. Narkhede, S. Mahajan, P. Bartakke, and M. Sutaone, “Towards compressed and efficient CNN architectures via pruning,” *Discover Computing*, vol. 27, no. 1, Sep. 2024, doi: <https://doi.org/10.1007/s10791-024-09463-4>.
- [14] A. M. Shaikh, Y. Zhao, A. Kumar, M. Ali, and Y. Kang, “Efficient Bayesian CNN Model Compression using Bayes by Backprop and L1-Norm Regularization,” *Neural Processing Letters*, vol. 56, no. 2, Apr. 2024, doi: <https://doi.org/10.1007/s11063-024-11593-1>.
- [15] B. Verbruggen and V. Ginis, “Improving image classification of one-dimensional convolutional neural networks using Hilbert space-filling curves,” *Applied Intelligence*, vol. 53, no. 22, pp. 26655–26671, Aug. 2023, doi: <https://doi.org/10.1007/s10489-023-04945-2>.
- [16] L. Meneghetti, N. Demo, and Gianluigi Rozza, “A dimensionality reduction approach for convolutional neural networks,” *Applied intelligence*, vol. 53, no. 19, pp. 22818–22833, Jul. 2023, doi: <https://doi.org/10.1007/s10489-023-04730-1>.
- [17] Siripuri Divya, Bhaskar Adepu, and P. Kamakshi, “Image Enhancement and Classification of CIFAR-10 Using Convolutional Neural Networks,” *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Jan. 2022, doi: <https://doi.org/10.1109/icssit53264.2022.9716555>.
- [18] M. Blot, M. Cord, and N. Thome, “Maxmin convolutional neural networks for image classification,” *arXiv (Cornell University)*, Oct. 2016, doi: <https://doi.org/10.48550/arxiv.1610.07882>.
- [19] K. Jordan, “94% on CIFAR-10 in 3.29 Seconds on a Single GPU,” *arXiv (Cornell University)*, Mar. 2024, doi: <https://doi.org/10.48550/arxiv.2404.00498>.
- [20] R. C. Çalik and M. F. Demirci, “Cifar-10 Image Classification with Convolutional Neural Networks for Embedded Systems,” *IEEE Xplore*, Oct. 01, 2018. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8612873> (accessed Dec. 21, 2021).
- [21] S. Aslam and Ali Bou Nassif, “Deep learning based CIFAR-10 classification,” Feb. 2023, doi: <https://doi.org/10.1109/aset56582.2023.10180767>.