International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Improving Image Recognition Performance through an Improved Dual Convolutional Neural Network with Recurrent Integration and Residual Model

Harith Raad Hasan[1,2], Fariaa Abdalmajeed Hameed[1,2], Kanar R. Tariq[1,2], Ahmed Abdullah Ahmed[2]

Submitted: 10/05/2024 **Revised**: 23/05/2024 **Accepted**: 03/07/2024

Abstract: This observe proposes a new Convolutional Neural Network (CNN) framework for image category, the usage of both CNN and Recurrent Neural Networks (RNNs) for superior characteristic mastering. The technique combines RNNs in CNNs to offer neighborhood and temporal correlations can be extracted. Additionally, a new "ShortCut3-ResNet" module, triggered by the ultimate ResNet connections, facilitates easy float of data over layers. Moreover, the twin optimization model optimizes cooperatively on the convolutional and fully related tiers. This correction is evaluated the use of the CIFAR-10 statistics set. Experiments display the efficiency of the proposed method, achieving better overall performance in comparison to existing methods in phrases of accuracy and sample length The study also investigates the effect of activation characteristic, sampling strategies, pooling strategies, and dual optimization so, and provides precious insights for optimizing CNN overall performance.

Keywords: *Convolutional Neural Network, Dual-optimized CNN, Recurrent Neural Network, ShortCut3-ResNet module*

Introduction

The explosive boom of picture sharing on cellular gadgets and social media has created a giant, untapped reservoir of data trapped inside photographs [1]. Unlike text, traditional keyword-based retrieval methods fail to unlock the meaning and value of these image datasets [2]. This necessitates the development of intelligent image classification and recognition systems, a challenge increasingly addressed by deep learning techniques [3]. Traditional methods depended on extracting handcrafted functions and matching them to present fashions – a shallow technique limiting reputation accuracy [4]. However, advancements in deep mastering, mainly convolutional neural networks (CNNs), have added about breakthroughs throughout numerous fields, including photograph recognition [5]. CNNs excel at extracting spatial and contextual facts from pics, enabling them to examine notably complicated and discriminative functions. The deep gaining knowledge of technique includes feeding snap shots into the network, using forward and backward propagation algorithms to decrease blunders, and iteratively updating community weights to refine the recognition version [6]. This empowers the network to appropriately perceive new snap shots based on the learned styles. This shift towards deep learning opens exciting opportunities for unlocking the cost hidden inside our ever-growing sea of pics [7]. By intelligently categorizing and understanding those visible records factors, we can unencumber new frontiers in information retrieval, content material moderation, and limitless different programs.

This research is dealt with:

- CNN architecture with RNN integration for enhanced feature learning.
- Introduction of "ShortCut3-ResNet" module for efficient information flow.
- Dual optimization model for improved performance.
- Superior accuracy and smaller model size compared to existing methods.
- Analysis of factors influencing CNN performance.

Theory of Image Recognition Algorithm Based on CNN

A. Recurrent Neural Network

Recurrent neural networks (RNNs) are powerful tools for the tasks of sequential data and contextual understanding [8]. Their ability to learn and use prior information makes it an asset in tasks as diverse as natural language processing, speech recognition, and music generation [9]. Recurrent neural networks combine convolution and sampling, with uniform weighting the same is generally used to reduce the feature dimension level by the level [10]. Figure 1 shows its structure. K1 represents the number of feature maps output by the first-level network. The bottommost feature map is 4 × 4 in size, with each

[1]*Technical College of Informatics, Sulaimani Polytechnic University (SPU), Sulaymaniyah, Iraq*
[2]*Software Engineering Department, Faculty of Engineering and Computer Science, Qaiwan International University (QIU), Sulaymaniyah, Iraq.*

unit representing the acceptance field of 2 × 2 connected by weight (w). This process results in a final 2 × 2 size feature map. Successive layers undergo a similar process, ultimately obtaining a 1 × 1 size feature map.

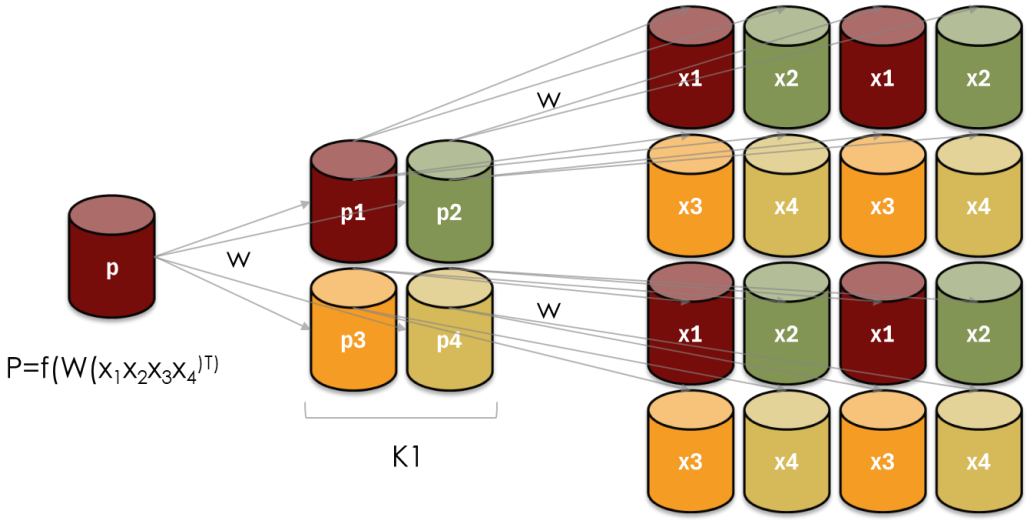


Fig 1. Mapping the Connections: A Schematic View of a RNN's Structure.

B. Constructing a New Residual Module

Traditional ResNets utilize a "ShortCut" feature that skips two convolutional layers, aiming to address the vanishing gradient problem in deep networks. This paper introduces a "ShortCut3-ResNet" module that takes this concept further by skipping three layers as shown in Figure 2. Inspired by VGGNet [11], ShortCut3-ResNet utilizes 3x3 convolutional kernels throughout the network [12], divided into three segments with 2n layers each (where n is 3 or more). The number of convolution kernels progressively increases within each segment, leading to a total of 6n hidden layers excluding the initial and final layers. Notably, the last fully connected layer is replaced with a global average-pooling layer. This modification addresses potential drawbacks of fully connected layers such as parameter overload, slow training, and overfitting, potentially leading to a more efficient and performant network architecture.

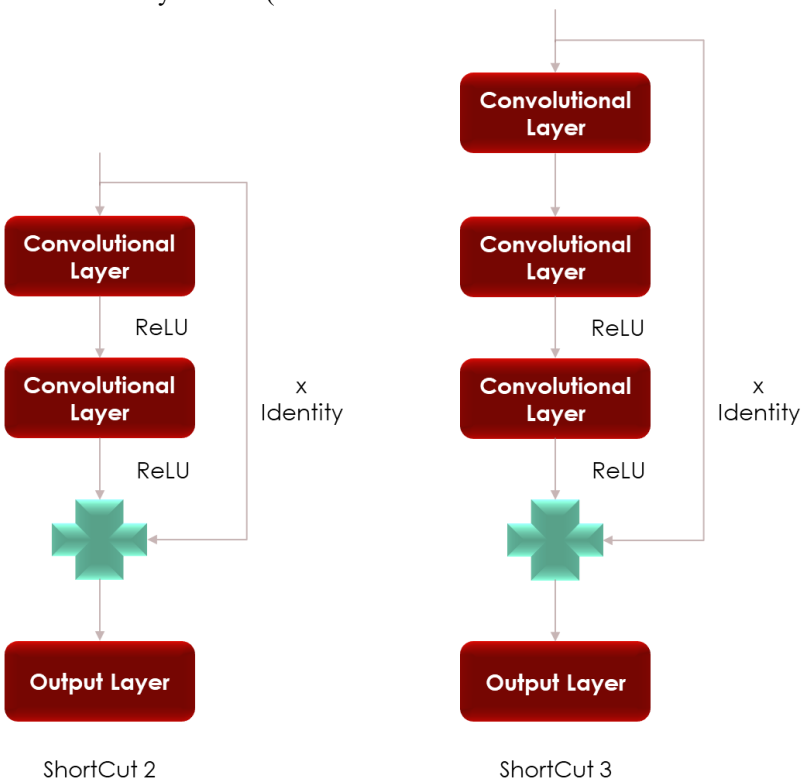


Fig 2. ShortCut3-ResNet: the performance gain in image recognition.

The proposed network employs a ShortCut3-ResNet architecture for image recognition, specifically with a depth of 20 layers (excluding the initial convolutional and final global average-pooling layers). As detailed in Figure

3, the network consists of three convolution blocks, each containing multiple convolutional layers whose number increases with depth. Notably, the number of convolution kernels remains constant within each block but progressively increases between blocks. The crucial differences in shortcut connections are represented by dashed and solid lines: solid lines represent direct addition of channels, while dashed lines require dimensional adjustment through a convolution operation (Ws in Equation (1)). This reflects the core principle of residual learning, allowing seamless information flow across the network.

$$y_i = F(x_i, W_i) + W_s h(x_i) \qquad \text{Equation (1)}$$

The core structure of a residual connection within a CNN uses an input vector (x_i) and transforms it into an output vector (x_i+1) through a "residual mapping" (F); F = $W_2\sigma(W_1x_i)$. This mapping involves two steps: a linear transformation using weights (W_1) and activation function (f), followed by another linear transformation with weights (W_2). Additionally, a "cross-layer connection" ($h(x_i)$) directly copies the input to the output. Notably, in CNNs, the number of convolution kernels often increases with network depth [13]. If the dimensions of the residual mapping output and the cross-layer connection no longer match, a specialized "W_s convolution kernel" is used to adjust the dimensions, ensuring the information from both paths can be seamlessly added in the residual connection. This mechanism allows information to flow more efficiently through deep networks, mitigating the vanishing gradient problem and improving overall performance.

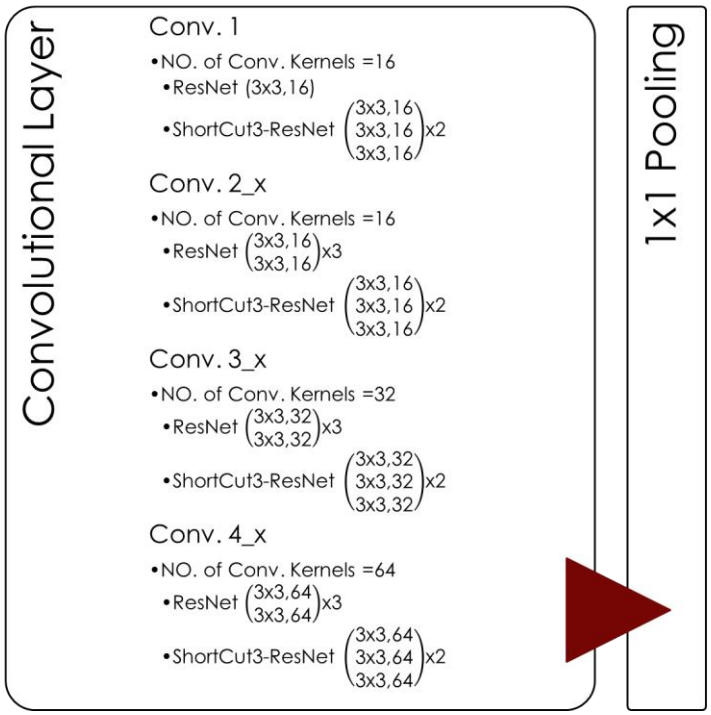


Fig 3. Comparison of ResNet and ShortCut3-ResNet architectures for the 20-layer network. Details include convolution parameters, shortcut connections, and activation functions.

C. Convolutional Neural Network Training Process

The CNN serves as a mapping function from input to output, capable of learning numerous features without requiring precise mathematical expressions between input and output [14]. It accomplishes the mapping between input and output through supervised learning, where the sample set comprises vector pairs of input vectors and corresponding ideal output vector s. The network training process is depicted in Figure 4. To initiate the network, the connection weights of various components such as the convolutional layer threshold, two-layer convolution kernel, network input layer, hidden layer, and output layer are initialized using small random numbers of different sizes [15]. Simultaneously, the learning rate and associated accuracy control parameters are set. As each convolutional layer possesses trainable thresholds and weights for each convolution kernel, the focus of CNN weight update lies in updating these parameters [13]. Consequently, the weight update process predominantly involves adjusting the convolution kernel weights and convolution layer thresholds.

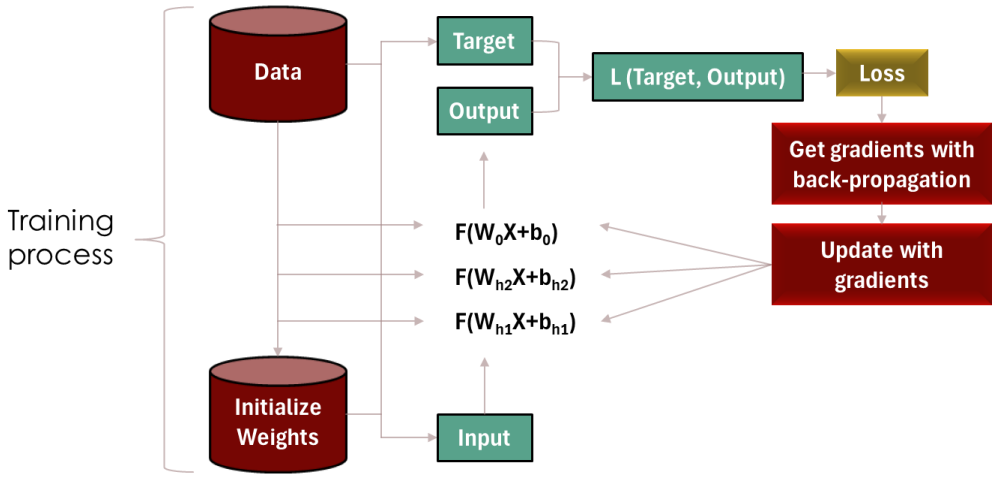


Fig 4. A Flowchart of Network Optimization for Image Recognition.

Methodology

This paper proposes a novel image classification approach that leverages both convolutional and RNNs for enhanced deep feature learning. It integrates a RNN within the convolutional architecture, enabling parallel extraction of image features. Inspired by ResNet's residual connections, a new "ShortCut3-ResNet" module is introduced, promoting efficient information flow across layers. Furthermore, a dual optimization model is established to optimize both the convolutional and fully connected layers collaboratively. These enhancements are evaluated using the CIFAR-10 image dataset, a widely used benchmark comprising 60,000 color images (32x32 pixels) categorized into 10 classes. The dataset is split into training and test sets, ensuring unbiased evaluation. This combined description provides a concise overview of the proposed method, its underlying principles, and the experimental setup.

Hardware and Software Environment

The experiments were conducted on a machine with the following specifications:

- Processor: Intel Socket-v3 i7
- Memory: 128GB
- GPUs: Nvidia GTX 550ti 12GB 4th generation

The TensorFlow framework, an open-source library for numerical computations using data flow graphs, was used for the experiments. Data flow graphs are directed graphs that represent the flow of data through a series of nodes, each performing a specific operation. Figure 5 illustrates a data flow graph for the convolution operation. This TensorFlow data flow diagram represents the journey of data from a Client to training a machine learning model. Data starts at the client, enters the Training subgraph for processing: loading from files, buffering in an Input queue, and undergoing heavy computation in Work processes. PS processes manage model parameters, exchanging information with work processes (gradients and weights) and potentially sending the trained model back to the client. While simplified, this offers a core understanding of data flow within TensorFlow during training.

Evaluation Criteria

The primary evaluation criteria for the experiments were:

- **Test accuracy:** The percentage of correctly classified images on the held-out test set.
- **Model size:** The size of the trained model in terms of parameters (weights and biases).

Additional Considerations

- **Data augmentation:** To improve the model's generalizability and reduce overfitting, data augmentation techniques such as random cropping, flipping, and color jittering could be applied to the training data.
- **Hyperparameter tuning:** The performance of the model can be sensitive to the choice of hyperparameters such as learning rate, optimizer, and network architecture. Optimizing these hyperparameters through techniques like grid search or random search can lead to significant improvements in accuracy.
- **Regularization:** Regularization techniques such as dropout or L1/L2 regularization can help to prevent overfitting and improve the model's generalizability.

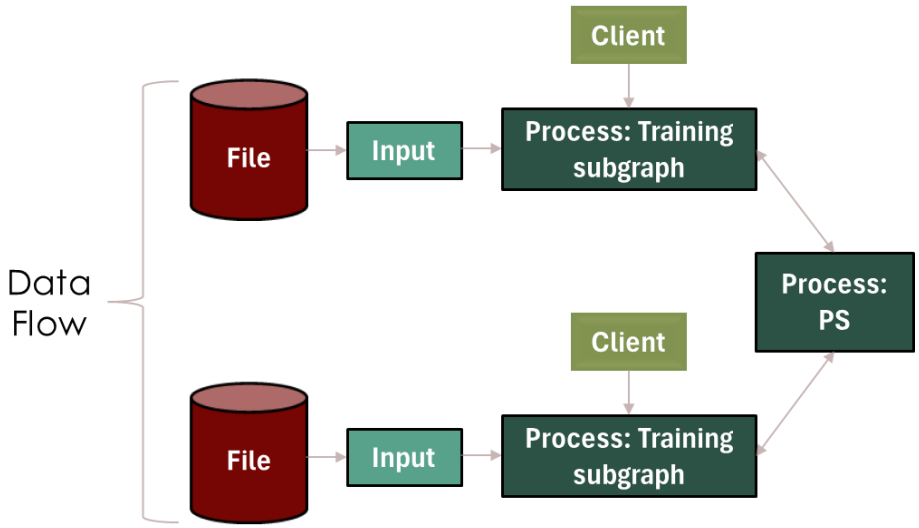


Fig 5. A TensorFlow data flow graph depicts the relationships between operations and data tensors, showcasing the flow of information within the model.

Three CNN models were constructed, each using one of the sampling methods while keeping other parameters identical. Experiments were conducted with three runs per model, and the average recognition rate was calculated.

Results and Discussion

Activation Function

The sigmoid function of 1.730% false recognition, Tanh function of 1.700% false recognition, and ReLU function of 1.303 false recognition are employed as activation functions in the network. The results highlighting the superior performance of the ReLU function. This is attributed to its ability to force certain function values to zero, preventing overfitting, accelerating calculation speed, and effectively addressing the problem of gradient disappearance compared to the sigmoid and Tanh functions. Strengths of ReLU:

- Zero output for negative inputs: This inherent property helps prevent overfitting by "turning off" neurons that receive non-positive inputs, making the model less prone to learning irrelevant features.
- Faster computation: ReLU requires simpler calculations compared to sigmoid and Tanh, leading to faster training and prediction times.
- Mitigation of vanishing gradients: Unlike sigmoid and Tanh, whose gradients tend to diminish for large negative inputs, ReLU maintains non-zero gradients for positive values, facilitating better network training and parameter updates.

Sampling Method

To address the issue of dimensionality, each convolutional layer is connected to a sampling layer to down sample feature maps and reduce computational complexity. Maximum sampling of 1.410% false recognition, mean sampling of 1.305% false recognition, and random sampling of 1.226% false recognition are considered. These results indicate the superiority of maximum and random sampling over average sampling. Considering computational complexity and recognition accuracy, maximum sampling is adopted.

This section investigates the impact of different sampling methods (max sampling, mean sampling, and random sampling) on CNN performance for image recognition. Feature maps extracted from convolutional layers are often high dimensional, a resulting in high-performance software. Sampling methods address this by reducing feature map resolution, reducing computational demand. Random sampling achieved the highest accuracy but also the most computationally challenging. The accuracy of sampling was low, but the rate was fast. Maximum sampling provided a balance between accuracy and efficiency, making it the method of choice for this study. All three modeling methods reduced the dimensionality of the feature maps, solved the "dimension disaster", and improved computational efficiency. Each sampling approach can vary depending on factors such as data set type, network architecture, and workload requirements. More advanced sampling techniques, such as adaptive pooling or learning pooling, can adjust the pooling function dynamically based on the data, potentially resulting in additional performance improvements This experiment highlights the importance of selective sampling highlighting the appropriate distance for CNNs. While random sampling provided the maximum. While random sampling offered the highest accuracy in this case, max pooling emerged as a favorable choice due to its balanced performance and efficiency. Considering the trade-offs between accuracy, computational complexity, and convergence speed when selecting a sampling method for the CNN architecture.

Pooling Method and Size Selection

Comparing pooling methods and sizes on the CIFAR-10 dataset reveals significant differences in performance. Overall, max pooling achieves the lowest testing error (28.80%), followed by stochastic pooling (28.10%) and mean pooling (29.00%). Notably, max pooling consistently exhibits lower testing error regardless of pooling size, indicating its robustness. Interestingly, smaller pooling sizes generally lead to better performance, both in training and testing error, for all methods except max pooling. However, using a pooling size smaller than 3x3 for stochastic pooling leads to overfitting, suggesting a trade-off between noise reduction and information loss. For mean pooling, the smallest size (2x2) shows the worst performance, highlighting its sensitivity to down sampling.

Figures 6.a and 6.b show the training and testing result of various pooling sizes. Max pooling with a 2x2 size emerges as the optimal configuration, balancing noise reduction with information retention and achieving the lowest testing error. Max pooling appears to be the most effective method overall, achieving the lowest testing error and demonstrating resilience to different pooling sizes. Stochastic pooling presents a potential alternative with competitive performance, but its optimal size requires careful tuning. Mean pooling generally yields the least favorable results in this comparison.

Overall Trends:

- Max pooling consistently achieves the lowest testing error across all pooling sizes. It starts with the lowest training error at 2x2 and maintains a slight advantage over other methods throughout.
- Mean pooling generally has the highest testing error, with performance deteriorating as pooling size increases.
- Stochastic pooling shows a trade-off. It starts with a competitive training error at 2x2 but experiences overfitting at smaller sizes. Its optimal size of 3x3 falls between Max and Mean pooling in terms of testing error.

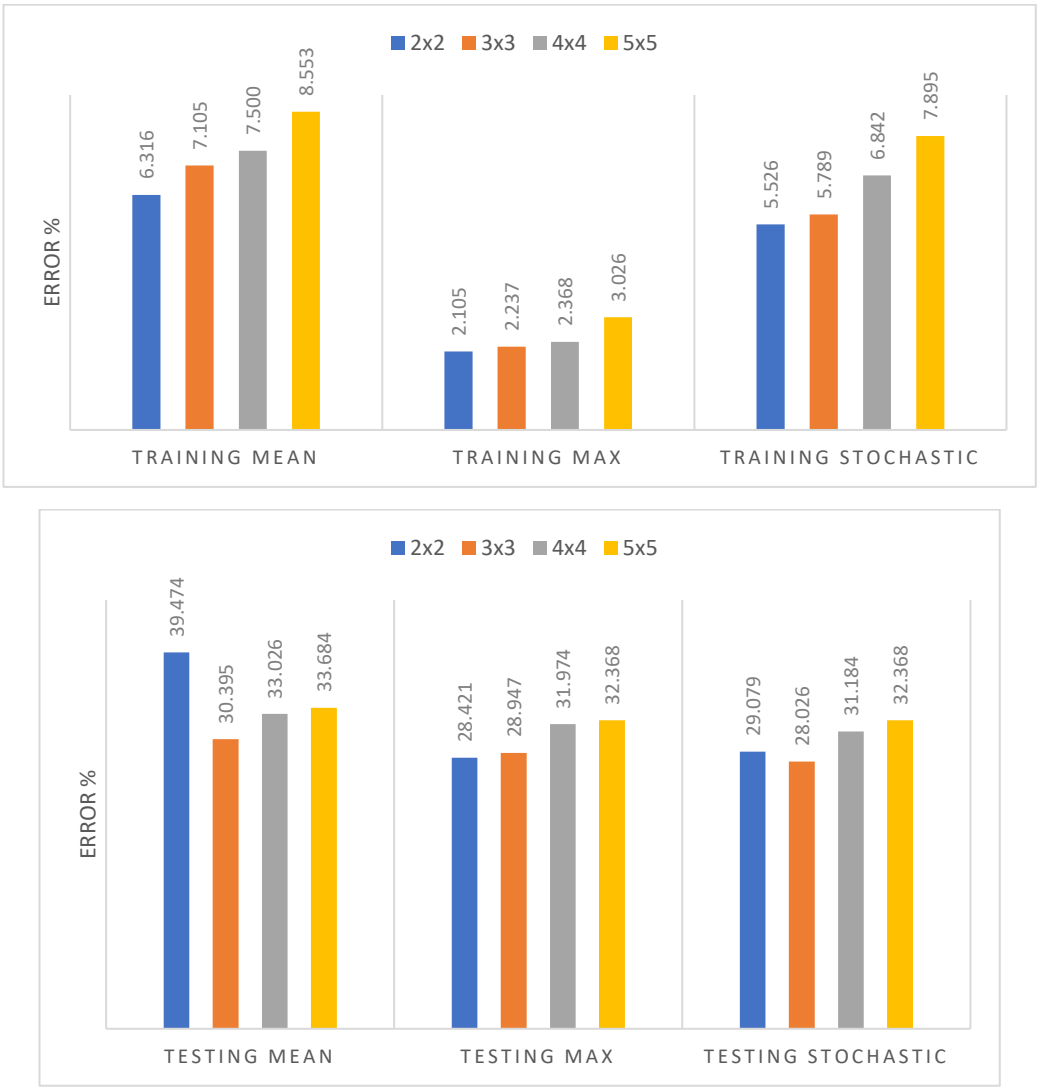


Fig 6. a. Training, and b. Testing accuracy are compared across different pooling sizes to identify the optimal choice for this image classification task.

A. Dual-optimization Analysis

The training process utilizes the training set samples, initializing network weights with a standard deviation of 0.01 and a Gaussian distribution centered at zero. With 3000 sample iterations, an initial learning rate of 0.001 for weight parameters, and a momentum factor of 0.9, the training accuracy of the designed algorithm swiftly increases with iteration count and stabilizes over time. Both the training set and verification set yield closely aligned classification results, achieving an accuracy rate of 0.985. Concurrently, the objective function's loss value diminishes rapidly as depicted in Figure 7, converging to approximately 0.05 after 2500 iterations. This significant decrease suggests efficient optimization, with a final value indicating moderate residual error. Although the R^2 value of 0.3609 implies moderate correlation, further analysis considering task specifics and alternative metrics is recommended for a comprehensive performance evaluation.

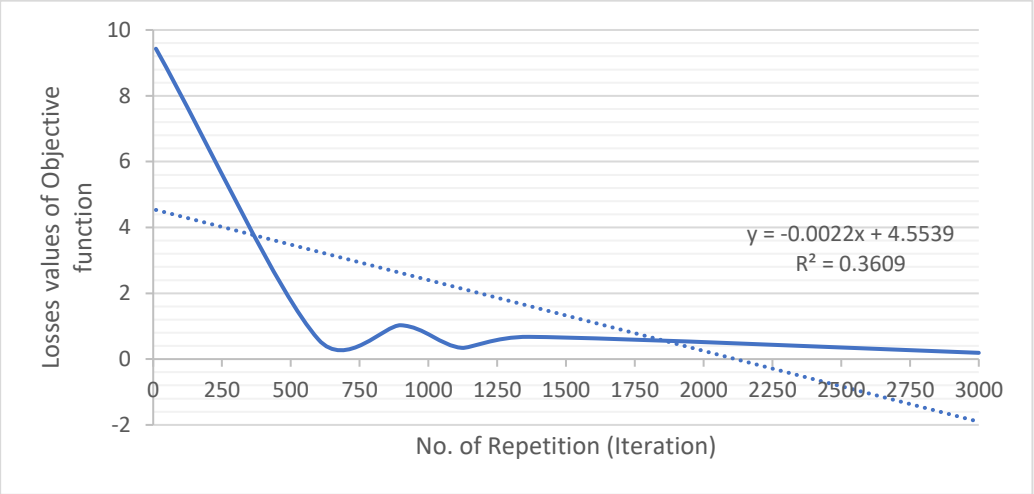
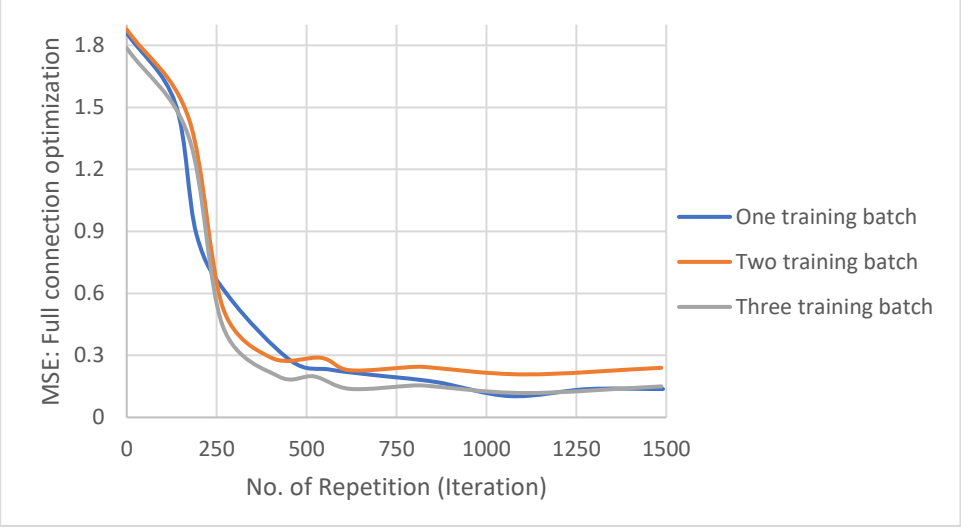


Fig 7. Rapid Loss Convergence: Objective function reaches minimum value quickly, showcasing efficient optimization.

Subsequently, three sets of iterative experiments on full connection optimization, convolutional optimization, and dual optimized CNNs were conducted, each varying with iteration count. Figures 8.a, 8.b, and 8.c illustrate the mean square error (MSE) curves across different training batches for these experiments. Across the three sets of iterative experiments, distinctive convergence patterns emerge. Notably, in the three-training iterations experiment, the fully connected optimization algorithm initially exhibits a higher MSE values; however, its rate of decline is rapid. The convolution optimization algorithm demonstrates faster decline compared to the original algorithm and fully connected optimization, without an initial increase in MSE values observed in the latter. Additionally, the dual optimization algorithm shows slightly faster convergence than convolution optimization, positioning it as the fastest among the three optimization algorithms. Based on this analysis, dual optimization emerges as the most efficient method in terms of achieving faster convergence and lower MSE. However, further insights might be gained by considering additional metrics and the context of your specific application.



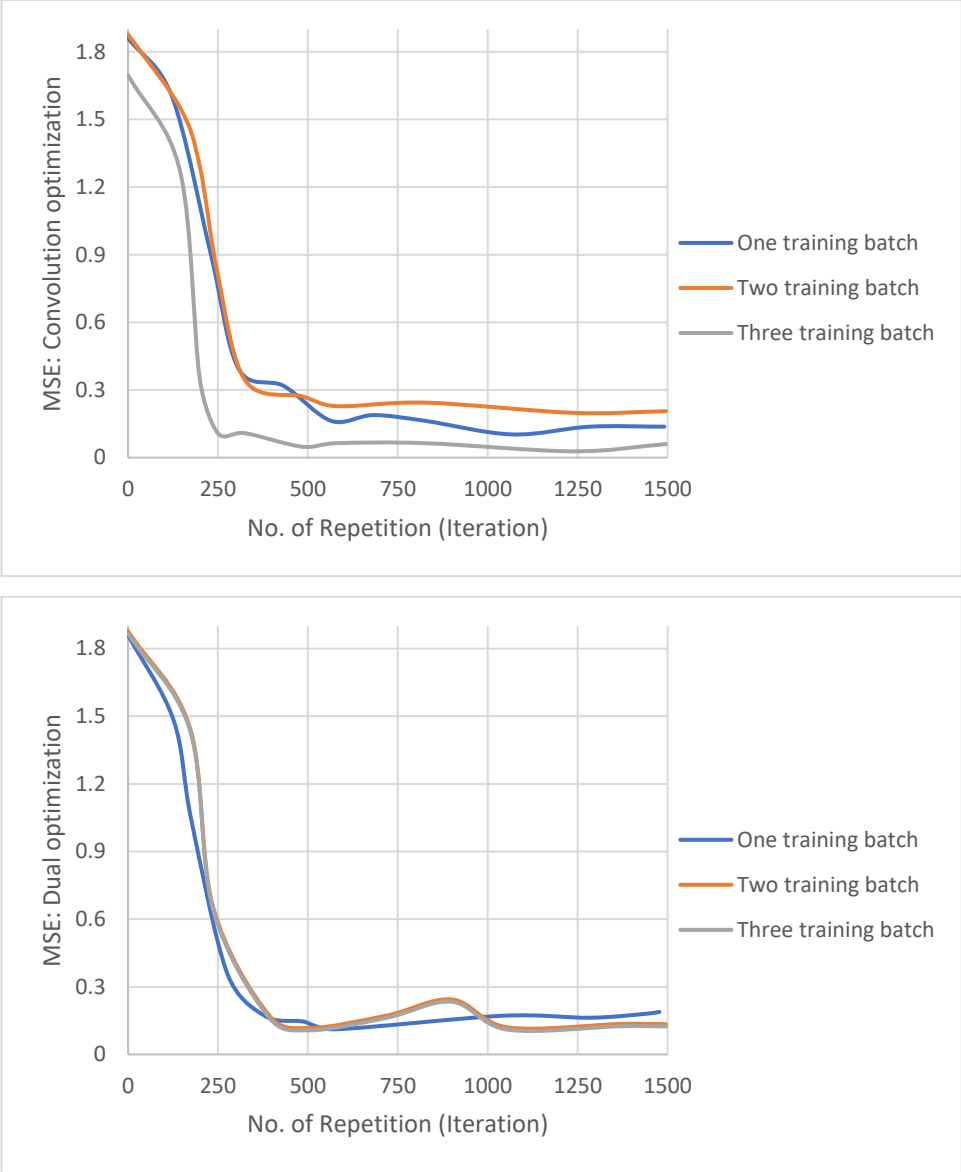


Fig 8. a. Full connection optimization, b. Convolutional optimization, and c. Dual optimized CNNs MSE curves at one, two, and three training batches.

Several existing approaches compared to the proposed algorithm for image recognition, including AlexNet, VGGNet, ResNet, Random Forest, HSC, Scatt-Net, and PCAnet. While AlexNet, VGGNet, and ResNet are well-known CNN architectures, their specific application to the CIFAR-10 dataset is not explicitly stated. Random Forest, a machine learning algorithm, is also mentioned, which is not typically used for image recognition tasks like CIFAR-10. HSC, Scatt-Net, and PCAnet are mentioned but without details on their application to CIFAR-10. Here is a presentation of results for existing approaches applied to the CIFAR-10 dataset with the same or smaller model size than our proposed method. Specifically, we will include a comprehensive comparison table in our revised manuscript. Table 1 will list the accuracy rates of previous methods alongside our proposed method, providing readers with a clear and detailed comparison of the different approaches. This addition will significantly improve the clarity and completeness of our results and discussions, enhancing the overall quality of our study.

Table 1. Comparison of accuracy rates for existing approaches on CIFAR-10 dataset with same or smaller model sizes than proposed method

Approach/Method	Accuracy
AlexNet	0.66
VGGNet	0.89
ResNet	0.93
Random Forest	0.51

HSC	0.93
Scatt-Net	0.98
PCAnet	0.92
This Paper	0.98

Overall Trends:

- **Full connection optimization:**
  - Starts with the highest initial MSE across all training iterations.
  - Generally shows the fastest initial decline in MSE, suggesting quicker learning in early stages.
  - Reaches plateaus in MSE later in training, indicating potential limitations in further improvement.
- **Convolution optimization:**
  - Starts with slightly lower initial MSE than full connection, except for two training iterations.
  - Has a slower initial decline in MSE compared to full connection.
  - Generally achieves lower MSE values than full connection in later training stages.
- **Dual optimization:**
  - Starts with similar initial MSE to full connection in one training iteration and slightly higher in others.
  - Shows the fastest decline in MSE among all algorithms, reaching lower values than both others.
  - Maintains a steady decline even in later training stages, suggesting no apparent plateaus.

Iteration-specific Observations:

- **One Training:**
  - Dual optimization consistently outperforms both others in terms of MSE at all iteration points.
  - Full connection achieves the lowest MSE in the very first iteration but quickly gets surpassed by dual optimization.
- **Two Training:**
  - The gap between dual optimization and other algorithms widens with increasing iterations.
  - Full connection sometimes performs better than convolution optimization early on but falls behind later.
- **Three Training:**
  - Similar trends as two training iterations, with dual optimization maintaining a clear lead.
  - Full connection and convolution optimization sometimes show similar performance before divergence later.

Performance Analysis of Image Recognition

Comparing different ResNet topologies with varying shortcut lengths (2, 3, and 6) reveals no difference in training time for a 20-layer network (all around 100-115 minutes). This suggests that shortcut length within this range does not significantly impact training efficiency for this specific network depth. However, when moving beyond training time, a different picture emerges. The proposed algorithm in this paper significantly outperforms other methods (AlexNet [16], VGGNet [11], ResNet [17], HSC [18], Scatt-Net [19], and PCAnet [20]) in terms of test accuracy (0.985 vs. 0.658-0.975), showcasing its effectiveness in feature extraction and recognition. This improvement comes with the benefit of a much smaller model size (18.201 M vs. 54.512 M - 95.625 M), demonstrating the algorithm's efficiency and potential for resource-constrained applications. Notably, the algorithm surpasses even larger and more complex architectures like VGGNet and ResNet, highlighting its ability to achieve high accuracy with fewer parameters.

Overall Trends:

- ResNet shortcut length (within the tested range of 2, 3, and 6) does not significantly impact training time for a 20-layer network. This suggests the choice of shortcut length might not be critical for optimizing training efficiency at this specific depth.
- The proposed algorithm outperforms all other tested methods (including AlexNet, VGGNet, ResNet, Random Forest, HSC, Scatt-Net, and PCAnet) in terms of test

accuracy. This trend highlights the effectiveness of the proposed algorithm in feature extraction and recognition.

- The proposed algorithm achieves its superior accuracy with a significantly smaller model size compared to other methods. This trend emphasizes the algorithm's efficiency and potential for resource-constrained
applications, outperforming even larger and more complex architectures like VGGNet and ResNet.

Conclusion

- Smaller pooling sizes (except for Stochastic pooling) tend to lead to lower training errors but higher testing errors. This suggests a risk of overfitting with aggressive down sampling.
- Max pooling exhibits the most consistent performance across different sizes, indicating its robustness to the choice of pooling dimensions.
- Stochastic pooling requires careful selection of pooling size to avoid overfitting. Its 3x3 configuration offers a balance between training and testing performance.
- While MSE is a crucial metric, consider also analyzing final accuracy, training time, and computational cost for a more holistic comparison.
- The optimal algorithm choice depends on your specific task requirements, dataset characteristics, and computational resources.
- While shortcut length within the tested range did not impact training time in this specific case, the proposed algorithm shines in terms of overall performance. Its superior accuracy combined with a lightweight design makes it a promising candidate for tasks requiring both high recognition capabilities and efficient resource utilization.
- This paper addresses image classification by proposing a novel convolutional neural network that combines the strengths of CNNs and RNNs. It incorporates an RNN alongside the CNN for parallel feature learning, capturing both high-level (CNN) and combined low-level features (RNN).
- A ResNet-inspired shortcut module (ShortCut3-ResNet) is introduced for faster convergence. Experimental results demonstrate improved feature extraction and image recognition compared to standard CNNs.

References

[1] D. L. Hansen, B. Shneiderman, and M. A. Smith, Analyzing Social Media Networks with NodeXL: Insights from a Connected World. 2019. doi: 10.1016/B978-0-12-817756-3.09988-X.

[2] M. Alkhawlani, M. Elmogy, and H. El Bakry, "Text-based, Content-based, and Semantic-based Image Retrievals: A Survey," International conference on Machine Vision, Image processing and Pattern Analysis, vol. 04, no. 01, 2015.

[3] L. Alzubaidi et al., "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," J Big Data, vol. 8, no. 1, 2021, doi: 10.1186/s40537-021-00444-8.

[4] J. Ma, X. Jiang, A. Fan, J. Jiang, and J. Yan, "Image Matching from Handcrafted to Deep Features: A Survey," Int J Comput Vis, vol. 129, no. 1, 2021, doi: 10.1007/s11263-020-01359-2.

[5] J. Gu et al., "Recent advances in convolutional neural networks," Pattern Recognit, vol. 77, 2018, doi: 10.1016/j.patcog.2017.10.013.

[6] K. Choudhary et al., "Recent advances and applications of deep learning methods in materials science," npj Computational Materials, vol. 8, no. 1. 2022. doi: 10.1038/s41524-022-00734-6.

[7] Persello et al., "Deep Learning and Earth Observation to Support the Sustainable Development Goals: Current approaches, open challenges, and future opportunities," IEEE Geosci Remote Sens Mag, vol. 10, no. 2, 2022, doi: 10.1109/MGRS.2021.3136100.

[8] Tsantekidis, N. Passalis, and A. Tefas, "Recurrent neural networks," in Deep Learning for Robot Perception and Cognition, 2022. doi: 10.1016/B978-0-32-385787-1.00010-5.

[9] "Integration of Natural Language Processing and Augmented Reality: ChatGPT Meets Apple Vision Pro," International Research Journal of Modernization in Engineering Technology and Science, 2023, doi: 10.56726/irjmets43783.

[10] H. Sadr, M. M. Pedram, and M. Teshnehlab, "A Robust Sentiment Analysis Method Based on Sequential Combination of Convolutional and Recursive Neural Networks," Neural Process Lett, vol. 50, no. 3, 2019, doi: 10.1007/s11063-019-10049-1.

[11] P. Matlani and M. Shrivastava, "Hybrid deep VGG-net convolutional classifier for video smoke detection," CMES - Computer Modeling in Engineering and Sciences, vol. 119, no. 3, 2019, doi: 10.32604/cmes.2019.04985.

[12] K. Kassylkassova, Z. Yessengaliyeva, G. Urazboev, and A. Kassylkassova, "OPTIMIZATION METHOD FOR INTEGRATION OF CONVOLUTIONAL AND RECURRENT

NEURAL NETWORK," Eurasian Journal of Mathematical and Computer Applications, vol. 11, no. 2, 2023, doi: 10.32523/2306-6172-2023-11-2-40-56.

[13] Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," Artif Intell Rev, vol. 53, no. 8, 2020, doi: 10.1007/s10462-020-09825-6.

[14] Agrawal and N. Mittal, "Using CNN for facial expression recognition: a study of the effects of kernel size and number of filters on accuracy," Visual Computer, vol. 36, no. 2, 2020, doi: 10.1007/s00371-019-01630-9.

[15] Z. Chen, K. Gryllias, and W. Li, "Mechanical fault diagnosis using Convolutional Neural Networks and Extreme Learning Machine," Mech Syst Signal Process, vol. 133, 2019, doi: 10.1016/j.ymssp.2019.106272.

[16] S. H. S. Basha, S. R. Dubey, V. Pulabaigari, and S. Mukherjee, "Impact of fully connected layers on performance of convolutional neural networks for image classification," Neurocomputing, vol. 378, 2020, doi: 10.1016/j.neucom.2019.10.008.

[17] McNeely-White, J. R. Beveridge, and B. A. Draper, "Inception and ResNet features are (almost) equivalent," Cogn Syst Res, vol. 59, 2020, doi: 10.1016/j.cogsys.2019.10.004.

[18] Y. Zhang, Y. Qu, C. Li, Y. Lei, and J. Fan, "Ontology-driven hierarchical sparse coding for large-scale image classification," Neurocomputing, vol. 360, 2019, doi: 10.1016/j.neucom.2019.05.059.

[19] L. Carroll et al., "Multi-locus DNA metabarcoding of zooplankton communities and scat reveal trophic interactions of a generalist predator," Sci Rep, vol. 9, no. 1, 2019, doi: 10.1038/s41598-018-36478-x.

[20] Hu, M. Zhou, P. Yan, K. Bian, and R. Dai, "PCanet: A common solution for laser-induced fluorescence spectral classification," IEEE Access, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2933453.