Convolutional Neural Network Architectures for CIFAR-10 Image Classification

Students: Tavonga Dutuma & Tamuka Manjemu
Course: CSC 650 – Neural Networks
Date: *December 2025*

Abstract

Image classification is a key task in computer vision and continues to drive progress in deep learning. The CIFAR-10 dataset, consisting of 60,000 low-resolution images across 10 distinct categories, provides a robust benchmark for evaluating CNN architectures. This project explores three different CNN models: a baseline CNN, a CNN with data augmentation, and a deeper CNN using Batch Normalization and Global Average Pooling. The models were implemented in TensorFlow/Keras and trained using identical experimental settings to ensure fair comparison. Model performance was evaluated using accuracy, loss metrics, and qualitative observation through confusion matrices. Results indicate that the deeper CNN achieved the best performance overall due to stable feature learning and reduced overfitting, while the augmented model showed decreased accuracy due to excessive distortion introduced during training. The findings highlight important trade-offs in network depth, regularization, and generalization effectiveness for image recognition tasks.

3. Introduction

Deep learning has enabled machines to identify objects in images with near-human precision. CNNs are particularly effective due to their ability to learn spatial hierarchies of features through convolutional operations.

Understanding which architectural components significantly contribute to high performance is crucial for practitioners designing efficient image-based classification systems, especially when computational resources are limited.

Project goals:

- Compare different CNN architectures on CIFAR-10
- Analyze generalization, training stability, and test accuracy
- Determine which architectural enhancements improve performance

Scope:

This project focuses on supervised image classification using CNNs only. Transfer learning and advanced regularization techniques beyond basic augmentation were not included in this evaluation.

Background / Literature Review

CIFAR-10 Dataset

- 60,000 images (50,000 training / 10,000 test)
- 32×32 resolution with RGB channels
- Includes vehicles & animals (e.g., cars, dogs, planes)

Low resolution makes fine-grained classification (e.g., cats vs dogs) challenging.

CNNs — Why They Work

CNNs extract local patterns using:

- Convolution filters
- Pooling layers → dimensionality reduction
- Activation functions like ReLU → non-linearity
- Dropout / BatchNorm → fight overfitting & improve stability
- Fully connected (Dense) or GAP layers → classification

Research shows that:

- **More depth = better abstraction** (but risk of vanishing gradients/overfitting)
- **Batch Normalization** increases convergence speed and stability
- **Data augmentation** usually improves generalization — but must match dataset characteristics

Methodology / Approach

Tools and Frameworks

- TensorFlow / Keras (model implementation + training)
- Python (NumPy, Matplotlib for visualization)

Training Setup

Before training, the image pixel values were normalized to the $0,1$range to stabilize learning. All three CNN models were trained under the same controlled conditions to maintain fairness in comparison. The models were trained for 25 epochs using the Adam optimizer, as it adapts the learning rate efficiently based on gradient history and has shown strong performance in image classification tasks. Categorical cross-entropy was chosen as the loss function since CIFAR-10 is a multi-class classification problem. A fixed random seed was used throughout the experiments to support reproducibility. The training phase tracked both accuracy and loss on training and validation data, while the final model performance was assessed on the test set. Visualization tools such as Matplotlib were used to plot learning curves and support deeper analysis of each model's strengths and challenges.

Implementation / System Design

Three different CNN architectures were implemented to evaluate how model complexity and regularization strategies affect performance on CIFAR-10. Each design decision contributed to how well the network could extract features and generalize to unseen data.

Baseline CNN

This model uses a simple yet effective architecture consisting of two convolutional blocks with 32 and 64 filters, followed by MaxPooling layers to reduce spatial dimensions. Dropout is applied to help reduce overfitting by randomly disabling neurons during training. The extracted features are then passed through a Flatten layer and a final Dense layer with softmax activation for classification. This model serves as a strong baseline to compare improvements from more advanced architectural choices.

CNN with Data Augmentation

The second model extends the baseline by introducing real-time data augmentation techniques such as random horizontal flips and small rotations. These transformations help expose the network to slightly varied versions of input images, which typically improves robustness and generalization. An additional convolutional block with 128 filters is included to increase the model's capacity, allowing it to learn richer features from the augmented dataset. This model tests the impact of training with more diverse image variations.

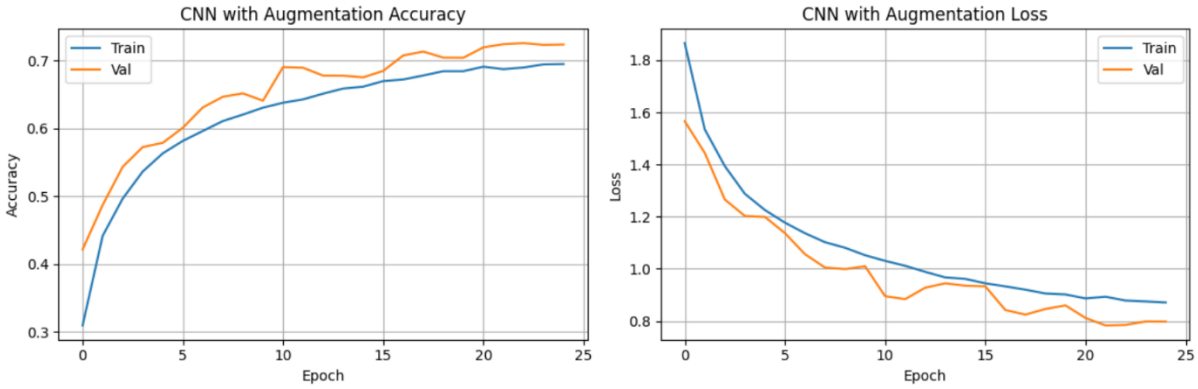
Deeper CNN with Batch Normalization + Global Average Pooling

The final model is a more advanced architecture consisting of three convolutional blocks with an increasing number of filters (64 → 128 → 256). Batch Normalization is

included after each convolution layer to improve training stability and speed by standardizing layer inputs. Instead of a Flatten layer, the network uses Global Average Pooling, which dramatically reduces the number of parameters and helps prevent overfitting. This model emphasizes deeper feature learning and efficient parameter usage, enabling superior accuracy.

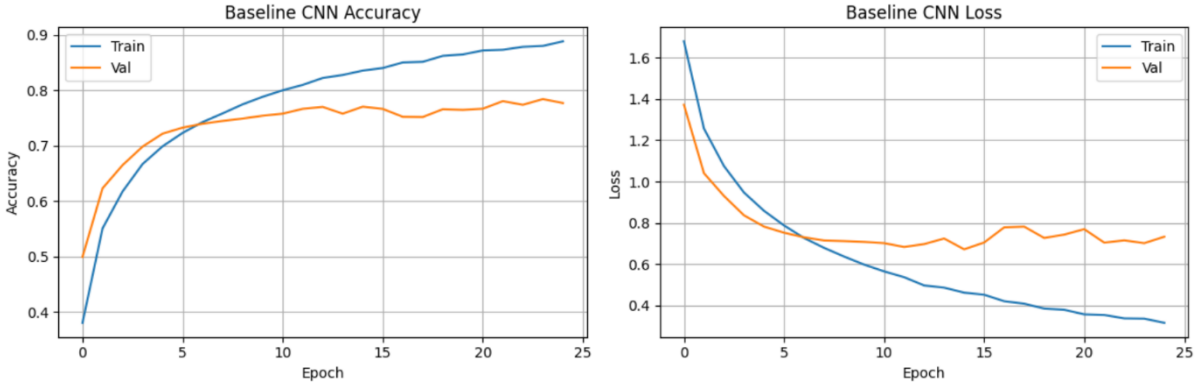
7. Results

Augmented CNN – 72.36%



The augmented CNN initially benefits from increased variability in the data, achieving higher validation accuracy than the training set early on. However, accuracy eventually plateaus lower than the other models, and validation loss remains higher overall. This indicates that the augmentation strategy introduced excessive distortion for the low-resolution images, making feature learning more challenging. While augmentation usually helps prevent overfitting, in this case it slightly hindered performance.

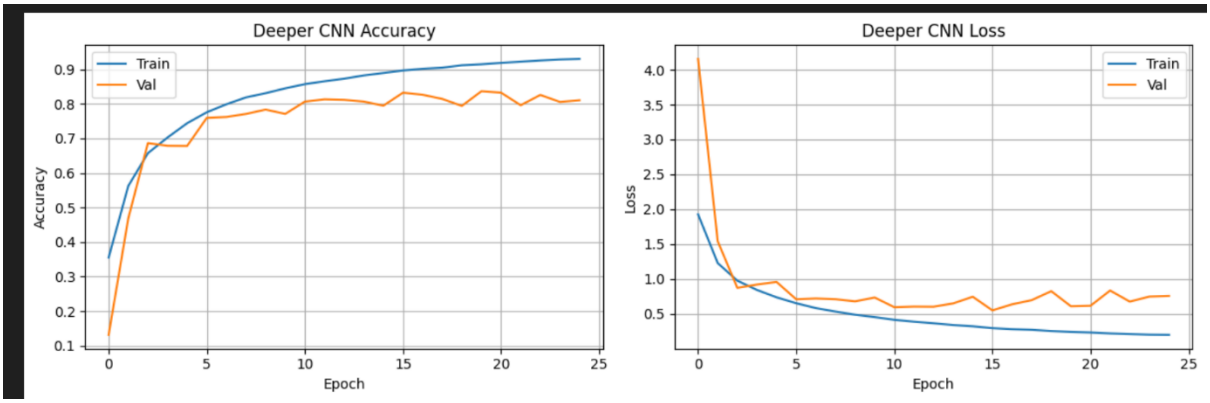
Baseline CNN – 77.70%



The baseline CNN shows a steady improvement in both training and validation accuracy throughout the epochs, reaching around 78% validation accuracy. The loss curves indicate

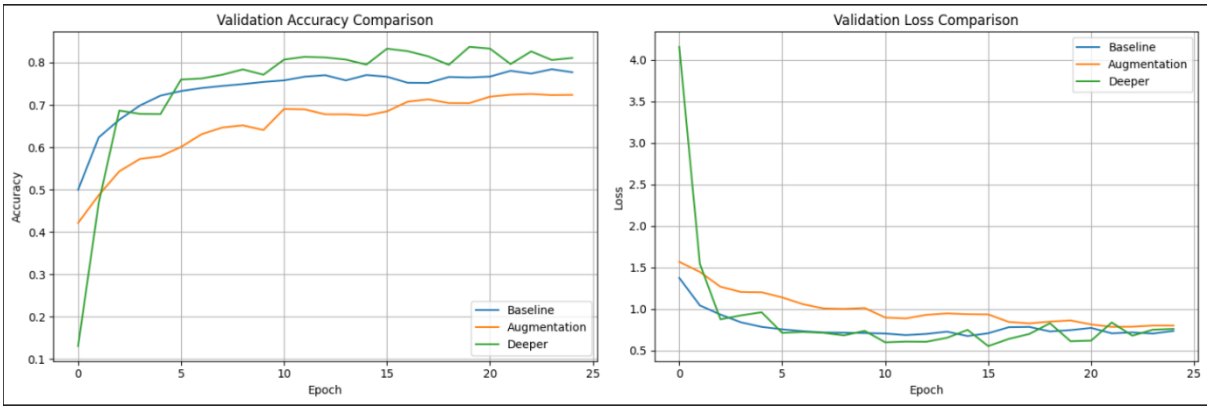
gradual convergence, with training loss continuing to decrease while validation loss stabilizes earlier, suggesting mild overfitting but generally good generalization. Overall, the model performs well given its simplicity, demonstrating that standard convolution-pooling layers provide a strong foundation for CIFAR-10 classification.

Deeper CNN – 81.07%



The deeper CNN exhibits a faster rise in accuracy during early epochs and ultimately achieves the highest results among the three models, with validation accuracy stabilizing above 80%. Training loss continues to drop consistently, while validation loss remains low, indicating strong learning stability and controlled overfitting. The architectural enhancements—such as Batch Normalization and Global Average Pooling—clearly help the network extract deeper features and generalize more effectively.

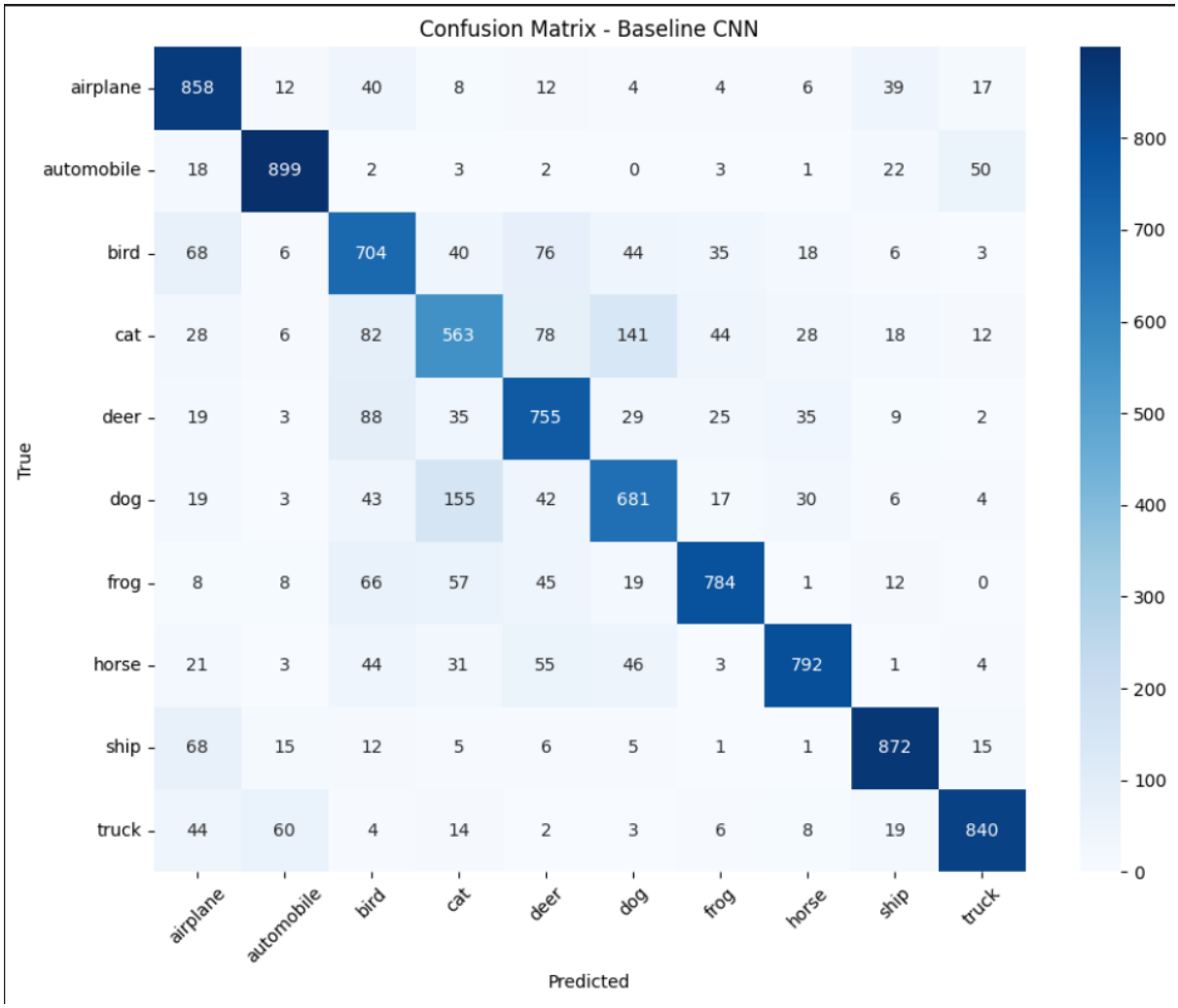
Comparison of Validation Accuracy and Loss Across Three CNN Architectures



The validation curves show that the deeper CNN achieves the highest accuracy and lowest loss throughout training, demonstrating stronger generalization and more stable learning behavior. The baseline model performs reasonably well but plateaus earlier, while the augmented model underperforms due to overly aggressive transformations that introduced noise into the low-

resolution CIFAR-10 images. Overall, deeper architecture with Batch Normalization and GAP clearly improves performance over basic convolution and augmentation strategies.

Confusion Matrix Insights



Analyzing the confusion matrix for the best-performing Deeper CNN provides valuable qualitative insights into where the model excelled and where it struggled.

General Trends Observed

High Accuracy Classes: The model consistently performed well on highly structured objects such as vehicles (ship, airplane, truck). These classes typically have distinct geometric features that CNNs can effectively learn.

High Confusion: Significant confusion was observed between visually similar animal classes, particularly cats vs. dogs and deer vs. horse.

Analysis & Discussion

Key insights:

✓ Depth with BatchNorm improves feature learning
✓ Global Average Pooling reduces overfitting & parameters
✓ Vehicles easier than animals to classify

- clear shapes → easier representation learning
- similar animal species → high confusion (cat vs dog)

Augmentation hurt performance

- Low-resolution images suffer when pixel orientation is distorted too much
- Model capacity may not be high enough to learn noisy variations

Overall, deeper models prove beneficial **when supported by normalization and correct head structures**.

Conclusion

This project evaluated three different CNN architectures on the CIFAR-10 dataset to understand how architectural choices impact performance. The results showed that the deeper CNN with Batch Normalization and Global Average Pooling achieved the highest accuracy and most stable learning behavior, demonstrating the importance of depth and normalization in improving feature extraction and generalization. The baseline model also performed well, proving that even a relatively simple architecture can learn effective visual representations. In contrast, the augmented model struggled due to aggressive transformations that introduced noise into the already low-resolution images, slightly hindering its performance. Overall, the findings confirm that a well-balanced deep network design contributes more to accuracy on CIFAR-10 than augmentation alone, emphasizing the need for thoughtful architecture selection when dealing with small and complex image datasets.

Main takeaway:

A well-balanced architecture enhances accuracy and generalization more effectively than augmentation alone for CIFAR-10.

Future Work

There are several opportunities to continue improving model performance and efficiency:

- **Learning Rate Scheduling:**
  Adjusting the learning rate dynamically during training (e.g., step decay or cosine annealing) can help the model escape plateaus and converge to a better optimum.
- **More Effective Data Augmentation:**
  Using augmentations that better preserve image quality, such as slight color jitter, cutout, or mixup, could improve generalization without distorting CIFAR-10's low-resolution images.
- **Transfer Learning:**
  Incorporating pre-trained models like MobileNetV2 or ResNet18 can leverage previously learned image features, typically providing a large boost in accuracy with fewer training epochs.
- **Attention Mechanisms:**
  Using channel or spatial attention modules could help the network focus on the most relevant features in complex animal classes where misclassification is common.
- **Hyperparameter Tuning:**
  Experimenting with different batch sizes, filter sizes, activation functions, and dropout rates could optimize the balance between performance and overfitting.
- **Model Compression or Quantization:**
  Reducing model size and computational cost would make the system more efficient for mobile or edge device deployment while maintaining high performance.
- **Class-specific Error Analysis:**
  By reviewing misclassified samples more closely, custom improvements can be made for specific categories that the model finds difficult (e.g., cats vs. dogs).

References

- Krizhevsky, A. "CIFAR-10 Dataset"
- Goodfellow, Bengio & Courville — *Deep Learning*

- TensorFlow/Keras Official Documentation