1ST MANDATORY ASSIGNMENT: NEURAL NETWORKS AND DEEP LEARNING

Georgios Lazaridis

November 2025

Aristotle University of Thessaloniki

1 Introduction and Implementation

This project implemented a **Feedforward Neural Network (FNN)** to solve the **Multi-Class Classification** problem on the **CIFAR-10** dataset. Two main architectures were tested: the initial **MLP (Multi-Layer Perceptron)** for compliance and the superior **CNN (Convolutional Neural Network)** for optimal performance.

1.1 Architectural Details

- **MLP Implementation:** Used **PCA (100 components)** to reduce the $32 \times 32 \times 3$ input dimension (3072) to 100 features before feeding into two Dense layers.
- **CNN Implementation:** Used **Conv2D** layers followed by **MaxPooling2D** layers to perform internal, learned feature extraction directly on the 32x32x3 image input.
- **Training:** Both architectures were trained using the **Back-propagation** algorithm via the `Adam` optimizer.

2 Comparative Experimentation and Results

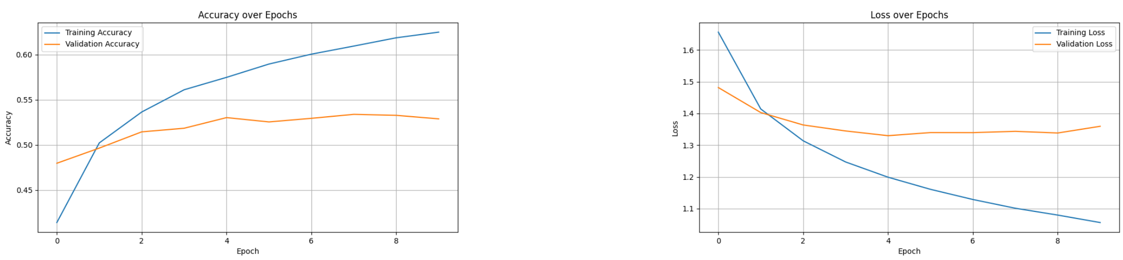
2.1 MLP (PCA) Performance Comparison

The MLP experiments focused on the impact of hidden neuron count (**H1/H2**) and epochs on performance, using PCA (100 components). The experiments using PCA achieved a peak Test Accuracy of 52.88%. However, the MLP(64/32) model (smallest capacity) performed best, suggesting larger models overfitted the limited PCA features. Furthermore, attempting to double the training time worsened the Test Accuracy, confirming severe overfitting across all MLP architectures.

Table 1: **MLP (PCA) Comparative Results**

	Architecture (H1/H2)	Epochs	Train Time (s)	Training Acc	Test Acc
	64 / 32	10	22.83	65.94%	**52.88%**
s	128 / 64	10	22.21	64.20%	51.95%
	256 / 128	10	22.56	64.71%	52.52%
	128 / 64	20	43.22	70.04%	51.75%

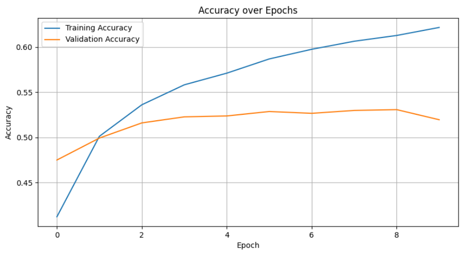
Figure 1: **Training Curves for MLP (64/32): Accuracy and Loss**



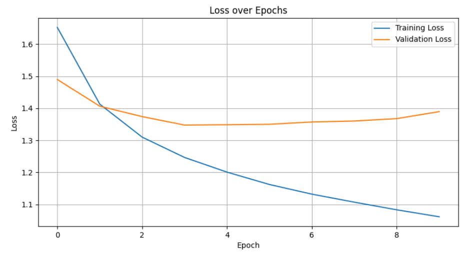
(a) MLP (64/32) Accuracy

(b) MLP (64/32) Loss

Figure 2: **Training Curves for MLP (128/64): Accuracy and Loss**

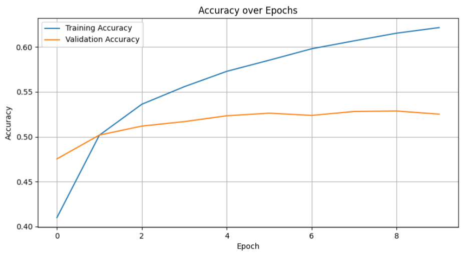


(a) MLP (128/64) Accuracy

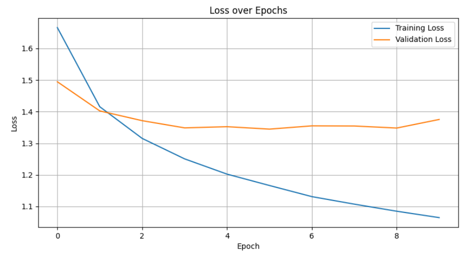


(b) MLP (128/64) Loss

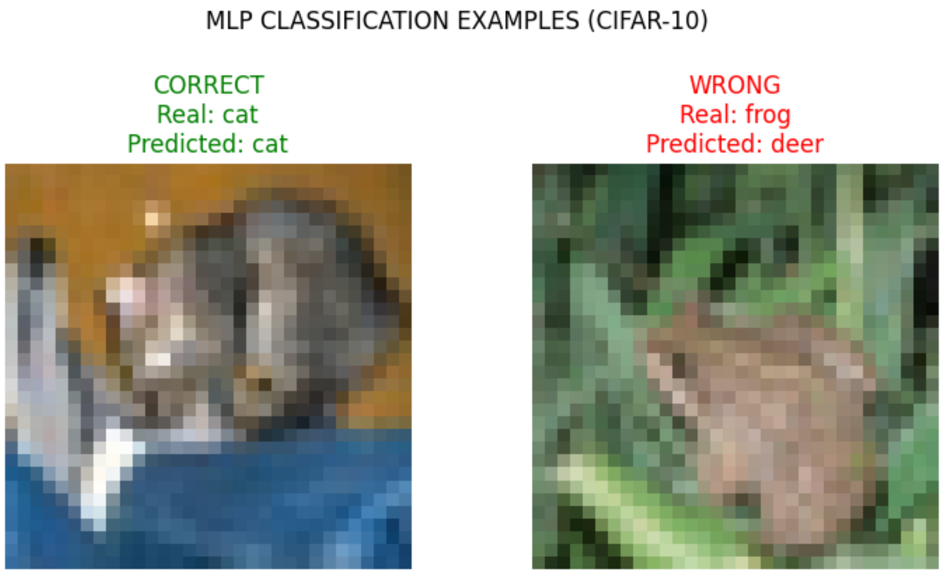
Figure 3: **Training Curves for MLP (256/128): Accuracy and Loss**



(a) MLP (256/128) Accuracy



(b) MLP (256/128) Loss



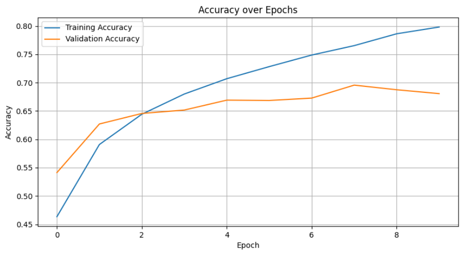
2.2 CNN Performance Comparison

The CNN experiments focused on the capacity of the convolutional layers (filters). The CNN architecture achieved significantly higher performance, peaking at 70.00% Test Accuracy with the CNN(64,128) model. The increased network capacity correlated positively with the Test Accuracy, as the largest model achieved the highest score. However, all CNN models displayed overfitting: training the CNN(32,64) for 20 epochs increased Training Accuracy to 95.87% but caused the Test Accuracy to drop (from 69.21% to 68.79%), confirming the network was memorizing noise after 10 epochs.

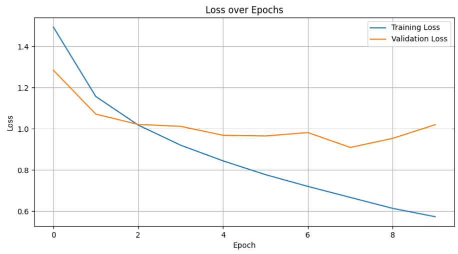
Table 2: **CNN Comparative Results**

Architecture	Epochs	Train Time (s)	Parameters	Training Acc	Test Acc
CNN(16, 32) - Dense128	10	187.41	153,962	81.57%	68.06%
CNN(32, 64) - Dense128	10	295.07	315,722	90.12%	69.21%
CNN(64, 128) - Dense128	10	704.47	666,890	**91.90%**	**70.00%**
CNN(32, 64) - Dense128	20	670.46	315,722	95.87%	68.79%

Figure 4: **Training Curves for CNN (16/32): Accuracy and Loss**

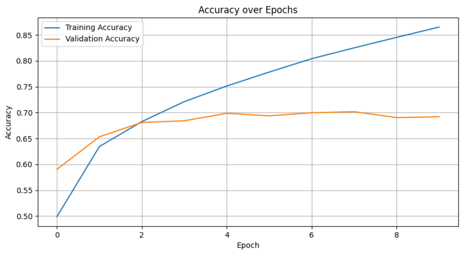


(a) CNN (16/32) Accuracy

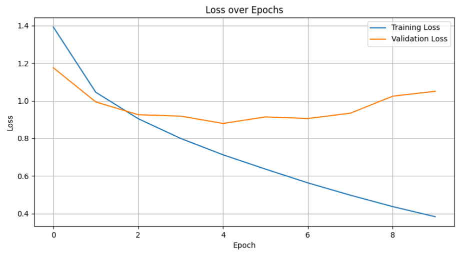


(b) CNN (16/32) Loss

Figure 5: **Training Curves for CNN (32/64): Accuracy and Loss**

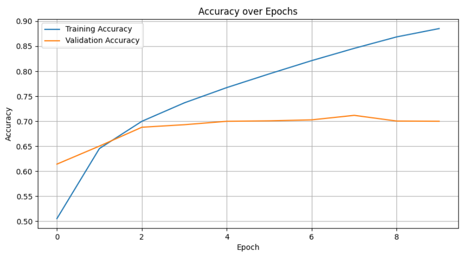


(a) CNN (32/64) Accuracy

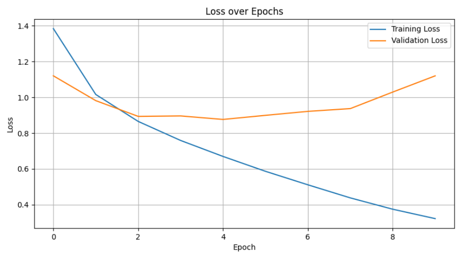


(b) CNN (32/64) Loss

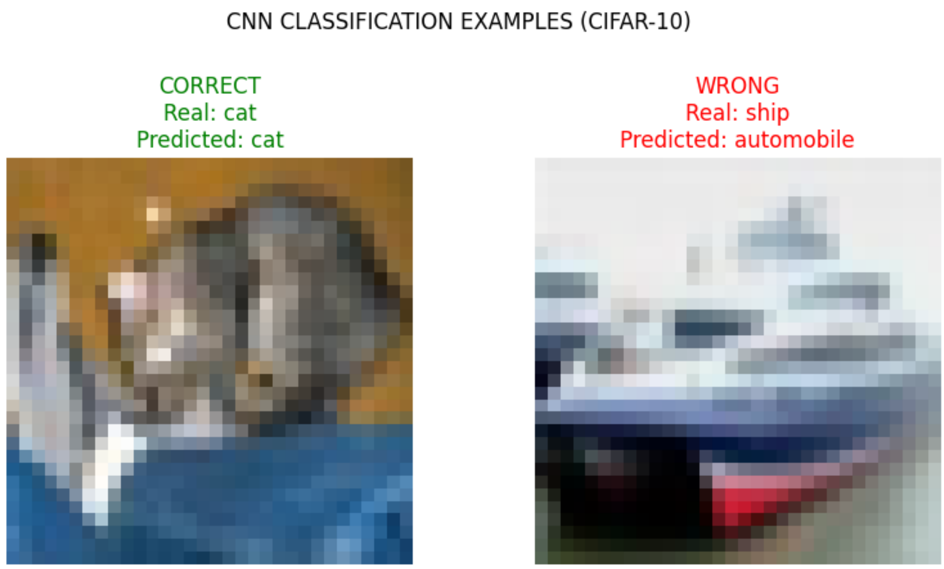
Figure 6: **Training Curves for CNN (64/128): Accuracy and Loss**



(a) CNN (64/128) Accuracy



(b) CNN (64/128) Loss



3 Analysis and Discussion

3.1 CNN vs. MLP vs. KNN vs. NCC

The CNN (70.00% Test Accuracy) achieved the highest performance by far, validating that learned feature extraction through Conv2D layers is essential for complex image classification tasks. The KNN (K=1) (37.38% Test Accuracy) and Nearest Centroid (27.75% Test Accuracy) models performed the poorest, even with PCA applied. This demonstrates that simple distance-based metrics are highly ineffective on this type of image data, as they cannot capture the non-linear, high-level features necessary for accurate classification.

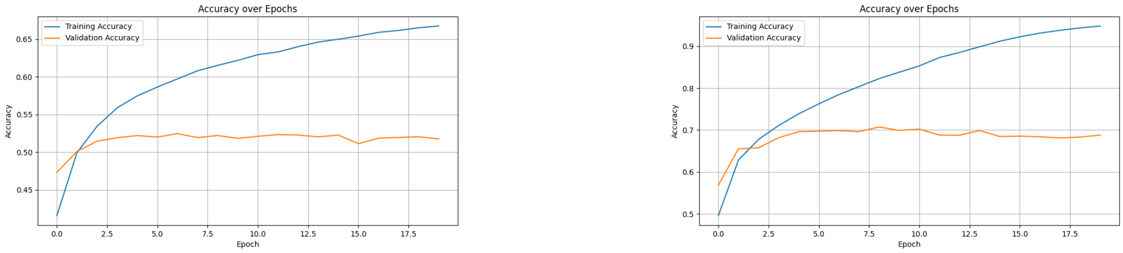
Table 3: **Final Peak Performance Comparison**

Model Type	Pre-processing	Peak Test Accuracy
CNN (64, 128)	None (Image Input)	**70.00%**
MLP (64/32)	PCA (100 Components)	52.88%
KNN (K=1)	PCA	37.38%
KNN (K=3)	PCA	35.34%
Nearest Centroid (NC)	PCA	27.75%

3.2 Overfitting and Epoch Limitation (Both Architectures)

- **MLP Overfitting:** In the MLP experiments, doubling the epochs (from 10 to 20) significantly increased Training Accuracy (from 64.20% to 70.04%) but caused the Test Accuracy to drop (from 51.95% to 51.75%), confirming overfitting.
- **CNN Overfitting:** The CNN displayed the same behavior: training for 20 epochs led to **95.87**% Training Accuracy but a drop in Test Accuracy (68.79%).
- **Conclusion:** For both FNN architectures, the networks fully converged and began to memorize noise after the first 10 epochs. This highlights the need for **Early Stopping** and **Dropout** regularization to improve generalization performance.

Figure 7: **Training Curves to Show Overfitting: Accuracy**



(a) MLP (128/64) Accuracy

(b) CNN (32/64) Accuracy

3.3 Code Implementation

The code uses the `Load or Fit` logic to ensure efficient experimentation. Visual evidence, including accuracy/loss curves and characteristic examples of correct and incorrect classification, was generated to fully document the network's behavior and support the findings. The complete codebase is available in the github repository: `https://github.com/lazoulios/neural-network-image-recognition`