ASSIGNMENT TOPIC- **Neural Network Models for Object Recognition Using Multi-Track ML Approaches**

1. TITLE SLIDE

This presentation explores the development and evaluation of neural network models for object recognition using the CIFAR-10 dataset. The project focuses on the application of Convolutional Neural Networks (CNNs) within the context of Track 2 Deep Learning under the theme *Neural Network Models for Object Recognition Using Multi-Track ML Approaches*.

The study examines data preparation, model architecture, training strategies, and evaluation metrics, while highlighting the advantages and trade-offs of using CNN-based approaches for visual object classification.

2. INTRODUCTION

Object recognition is a fundamental task in computer vision that enables machines to identify and categorize visual elements within images. It forms the basis of various real-world applications such as autonomous driving, medical imaging, and security systems.

In recent years, the integration of machine learning and deep learning has substantially improved recognition accuracy and computational efficiency (Sarkar *et al.*, 2024). Traditional machine learning approaches rely on manual feature extraction, where relevant image characteristics are identified before classification. In contrast, deep learning models such as Convolutional Neural Networks (CNNs) automatically learn these features directly from data, reducing the need for manual intervention.

This presentation focuses on exploring two neural network architectures applied to the CIFAR-10 dataset a widely used benchmark in image classification to evaluate their performance and highlight the advantages of deep learning for object recognition.

3. DATASET OVERVIEW

The CIFAR-10 dataset, developed by Krizhevsky *et al.* (2009), is a benchmark dataset widely used for image classification and object recognition tasks. It consists of **60,000 colour images**, each with a resolution of **32 × 32 × 3 pixels**, representing small, low-resolution real-world objects. The dataset is divided into **10 distinct classes**, including airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck.

For this project, the dataset was split into **50,000 images for training** and **10,000 for validation and testing**. To enhance model generalisation and ensure consistent input scaling, preprocessing steps such as **normalisation** and **one-hot encoding** were applied. Additionally, **data augmentation** techniques including horizontal flips, rotations, zooming, and shifting were used to expand the effective training data and prevent overfitting.

Figure 2 provides a visual representation of the dataset's classes, illustrating the diversity and complexity of the images used in training and evaluation.

4. DATASET PARTIONING

The dataset was divided into three subsets to ensure effective training and reliable evaluation of model performance: **40,000 images (66.7%) for training**, **10,000 images (16.7%) for validation**, and **10,000 images (16.7%) for testing**.

Stratified sampling was applied to maintain class balance across all subsets, ensuring that each of the ten object categories was equally represented. This is a crucial step for preventing model bias toward certain classes and promoting generalisable learning.

The **validation set** was specifically used to monitor model generalisation during training, helping to identify signs of overfitting and allowing for the fine-tuning of hyperparameters. This approach aligns with best practices in deep learning experiments, as outlined by Géron (2022).

The pie chart in Figure 3 visually summarises this data split, highlighting the proportion of samples allocated to each subset.

5. METADATA AND CLASS DISTRIBUTION

After partitioning the CIFAR-10 dataset into training, validation, and testing subsets, the key metadata was examined to confirm consistency and data quality. Each image retained a fixed resolution of **32 × 32 pixels** with **three colour channels (RGB)** and a **normalised pixel range between 0 and 1**.

The mean RGB intensity values were approximately balanced across channels, indicating uniform colour representation and image diversity. These statistics confirm that the dataset is well-prepared for training without introducing bias due to uneven lighting or contrast.

As illustrated in the bar chart, each class including *airplane, automobile, bird, cat, deer, dog, frog, horse, ship,* and *truck* contains an equal number of samples across the training, validation, and test splits. This class balance ensures that the model is trained and evaluated fairly, promoting unbiased generalisation and reliable performance outcomes.

6. NORMALISATION

Normalisation was applied to rescale all pixel intensity values in the CIFAR-10 dataset from the original range of **0–255** to a **0–1 scale**. This transformation ensures that all images are processed on a consistent numerical scale, improving training stability and preventing issues such as exploding or vanishing gradients.

As illustrated in the image grid, normalisation maintains the structural integrity and visual details of the original images, ensuring that only the pixel range changes not the image quality.

According to **Rawat and Wang (2017)**, applying normalisation accelerates model convergence, enhances feature extraction, and promotes smoother gradient updates during training. Overall, this preprocessing step contributes significantly to achieving more stable and efficient model learning.

7. NORMALISATION HISTOGRAM

The histogram comparison illustrates how pixel intensity distributions remain statistically consistent after normalisation. Prior to scaling, pixel values were spread between 0 and 255. After transformation, these values were compressed into the [0–1] range, while maintaining the same frequency distribution shape.

This confirms that the scaling operation preserved the dataset's inherent characteristics and visual balance. The consistent distribution across both histograms demonstrates that normalisation modifies only the numerical scale not the underlying data.

Consequently, this preprocessing step reduces computational cost, enhances numerical stability, and facilitates smoother gradient flow throughout the convolutional neural network.

8. ONE-HOT ENCODING

After completing image preprocessing, the class labels were encoded using the one-hot encoding technique. This approach transforms each categorical label within the CIFAR-10 dataset into a binary vector of length ten, in which a single “hot” index is assigned the value of one to denote the correct class.

This encoding method prevents ordinal bias by ensuring that all categories are treated as mutually exclusive and non-hierarchical. Additionally, it integrates effectively with the softmax activation function and categorical cross-entropy loss, enabling the network to generate class probability distributions during training.

As observed in the visual example, each image corresponds to a single “hot” value within its respective vector, reinforcing the distinctness of each class.

Zhu, Qiu, and Fu (2024) emphasise that one-hot encoding remains one of the most interpretable and dependable strategies in multi-class classification, as it provides a clear representation that facilitates efficient neural network learning and performance evaluation.

9. DATA AUGUMENTATION

Data augmentation was applied to artificially expand the diversity of the CIFAR-10 training set without collecting new data. A range of random transformations including rotation, horizontal flipping, and minor width and height shifts were performed solely on the training images to preserve the integrity of the validation and test sets.

These transformations introduce subtle variations in the dataset, enabling the convolutional neural network to learn invariant representations of objects across different orientations and spatial positions. By exposing the model to multiple visual perspectives of the same object class, augmentation reduces overfitting and improves generalisation to unseen data.

The augmented samples, illustrated in the figure, maintain their original semantic meaning while appearing slightly altered, thereby enriching the learning space.

Shorten and Khoshgoftaar (2019) affirm that such augmentation techniques enhance model robustness by simulating real-world variability and improving overall recognition accuracy in deep learning applications.

10. OVERVIEW OF CONVOLUTIONAL NEURAL NETWORK WORKFLOW

The Convolutional Neural Network (CNN) architecture operates through a series of hierarchical stages designed to extract, interpret, and classify visual information from input images. Initially, images are processed through convolutional layers that apply multiple filters or kernels to detect low-level patterns such as edges and textures. These representations are progressively refined through additional convolutional and pooling layers, resulting in the extraction of higher-level features.

Following feature extraction, the resulting feature maps are flattened and passed through fully connected layers, which perform classification by associating the learned features with specific output classes. The training process involves forward propagation where the model makes predictions and backpropagation where weights and kernels are iteratively adjusted to minimize the loss function.

This dynamic feedback loop allows the CNN to automatically learn complex image representations and achieve high recognition accuracy in datasets such as CIFAR-10. Yamashita et al. (2018) emphasise that this layered approach enables CNNs to capture both local and global image features efficiently, facilitating robust visual pattern recognition.

11. CNN ARCHITECTURE

The Convolutional Neural Network (CNN) architecture comprises a sequence of interconnected layers that progressively transform input images into abstract feature representations for classification.

Initially, convolutional layers apply multiple kernels to detect fundamental visual patterns, including edges and textures. These operations are followed by Rectified Linear Unit (ReLU) activation layers, which introduce non-linearity and enhance the model’s capacity to learn complex relationships within the data.

Pooling layers then reduce the spatial dimensions of the feature maps, preserving salient information while mitigating overfitting and computational cost. Subsequently, the extracted features are flattened and passed through fully connected layers that combine the learned representations to produce final class predictions.

The output layer employs a SoftMax activation function to generate class probabilities for categories such as airplane, dog, or car within the CIFAR-10 dataset.

This hierarchical design enables CNNs to automatically extract, refine, and interpret image features, resulting in efficient and accurate visual recognition performance (Rawat and Wang, 2017; Alzubaidi et al., 2024).

12. REASON FOR CHOSEN MODEL

This diagram illustrates the conceptual distinction between traditional machine learning and deep learning methodologies. In conventional machine learning workflows, the feature extraction process is performed manually before the classification stage. This requires prior domain knowledge and extensive feature engineering to identify patterns relevant to the task.

Deep learning, by contrast, integrates feature extraction and classification into a single, end-to-end process. Through hierarchical layers of representation learning, models such as Convolutional Neural Networks (CNNs) automatically discover the optimal features from raw input data.

This automation significantly enhances scalability and efficiency, particularly when working with high-dimensional image datasets such as CIFAR-10. The ability of CNNs to autonomously learn rich feature hierarchies makes them a superior choice for complex visual recognition tasks. Consequently, this project adopts the deep learning approach under Track 2 to leverage these advantages.

13. BASELINE CNN ARCHITECTURE

The baseline Convolutional Neural Network (CNN) architecture was designed as a lightweight yet effective model for image classification on the CIFAR-10 dataset. It comprises three convolutional–pooling blocks that extract progressively complex spatial features while reducing the dimensionality of the input images from 32×32 to 4×4 pixels.

Following feature extraction, the output is flattened and passed through two fully connected (dense) layers, culminating in a soft-max output layer responsible for classifying images into ten distinct categories. The model's total number of trainable parameters is approximately 356,810, making it computationally efficient for training while maintaining adequate representational power.

This architecture serves as the foundational model in the experimental workflow. It establishes a benchmark for evaluating the performance improvements achieved through later models that incorporate optimization and transfer learning strategies.

14. EPOCHS AND TRAINING STRATEGY

The baseline Convolutional Neural Network (CNN) was trained using the Adam optimizer, initialized with a learning rate of 0.001. Adam was selected for its adaptive learning capabilities, which facilitate faster convergence and improved stability during training.

The training process was configured for a maximum of 15 epochs with a batch size of 64, but the EarlyStopping mechanism halted training automatically at epoch 9 when the validation loss plateaued. This approach effectively mitigated overfitting and enhanced model generalization.

A validation split of 20% was applied to monitor model performance on unseen data. The ReduceLROnPlateau callback further optimized the training process by reducing the learning rate whenever the validation loss ceased to improve, allowing for more refined weight adjustments in later epochs.

Collectively, these techniques ensured efficient learning, controlled generalization, and established the baseline model as a reliable reference point for subsequent experimentation and optimization in Model 2.

15. TRAINING PERFORMANCE

In this slide, I present the training performance of the baseline convolutional neural network model.

The training accuracy improved steadily across epochs, reaching approximately **95%** by the final epoch.
Validation accuracy, on the other hand, plateaued around **73%**, which suggests that the model began to slightly overfit as training progressed.

In the loss plot, we can see that the **training loss decreased consistently**, while the **validation loss started to rise after epoch eight**. This confirms the same overfitting pattern observed in the accuracy curve.

Overall, the model performed well in learning from the training data, but its generalisation to unseen data was somewhat limited, a common trait of early CNN baselines before regularisation and optimisation are introduced.

16. EVALUATION METRICS

This table summarises the key evaluation metrics for the baseline CNN model.

The **training accuracy** reached **0.8753**, while the **validation** and **test accuracies** were **0.7358** and **0.7289**, respectively.
These results indicate that the model generalised fairly well, with consistent validation and test outcomes.

However, the **training loss** was lower (**0.366**) compared to the **validation loss (0.8529)** and **test loss (0.8782)**, suggesting mild overfitting as the model performed slightly better on the training data than on unseen data.

The model achieved its best performance at **epoch 8**, where early stopping was triggered to prevent further overfitting and preserve the optimal weights.
Overall, this performance establishes a solid foundation for comparison with the enhanced model introduced next.

17. CONFUSION MATRIX

This slide presents the evaluation results of the baseline CNN model on the CIFAR-10 test set. The model achieved a **test accuracy of approximately 73%**, which aligns with the validation accuracy and demonstrates good generalisation despite the model's simplicity.

From the **classification report**, we can see that performance varied slightly across classes. The model performed best on **automobile, ship, and truck**, each achieving precision and recall scores above 0.80 likely because these classes have more distinct visual features.
On the other hand, categories such as **cat, dog, and bird** were more challenging, with F1-scores around 0.55 to 0.65, mainly due to their visual similarities.

The **confusion matrix** reinforces these findings. The strong diagonal pattern shows correct predictions for most images, while the off-diagonal values particularly between cat and dog indicate areas of confusion.
Overall, these results show that the model learned to distinguish broad object categories effectively but struggled with fine-grained differences.

18. ENHANCED CNN ARCHITECTURE

This model represents the enhanced convolutional neural network developed to improve upon the baseline architecture. The structure incorporates additional convolutional layers, allowing for deeper hierarchical feature extraction.
Batch Normalisation layers were introduced after each convolutional layer to stabilise learning, reduce internal covariate shift, and enable faster convergence.
Dropout layers were strategically added to mitigate overfitting by randomly deactivating neurons during training.
The enhanced network maintains computational efficiency with approximately 530,000 trainable parameters, while achieving stronger generalisation and more stable learning dynamics compared to the baseline CNN.

19. EPOCHS AND TRAINING STRATEGY

For the enhanced CNN, the training strategy was extended and refined to improve convergence and model generalisation.
The Adam optimiser with an initial learning rate of 0.001 was maintained, supplemented by

ReduceLROnPlateau and *EarlyStopping* callbacks to adaptively regulate the learning process. The model was trained for 25 epochs with a batch size of 64, reaching optimal stability around the twentieth epoch.
A 20% validation split was preserved to monitor overfitting and tune performance.
These enhancements led to smoother convergence, reduced overfitting, and improved validation accuracy compared to the baseline CNN.

20. TRAINING PERFORMANCE

Model 2 demonstrated stronger and more stable learning dynamics compared to the baseline CNN.Training and validation accuracy increased consistently, converging around 0.78 to 0.80, with minimal divergence between the two curves.
Loss values decreased smoothly across epochs, supported by the integration of batch normalisation, dropout, and adaptive learning rate adjustments.
These enhancements contributed to improved model regularisation, reduced overfitting, and smoother convergence throughout training.

21. CONFUSION MATRIX AND INISGHTS

The confusion matrix for Model 2 demonstrates notable improvement in classification performance relative to the baseline CNN. The diagonal dominance indicates that the model correctly identified most test images within their respective categories, while off-diagonal elements were lighter, signifying fewer misclassifications.
Enhanced feature extraction through additional convolutional layers, combined with batch normalisation and dropout, contributed to improved stability and class separation.
This led to stronger predictive accuracy across all ten CIFAR-10 classes, particularly in categories with distinct visual features such as automobiles, ships, and trucks.

22. EVALUATION METRICS

Model 2 achieved superior evaluation metrics compared to the baseline CNN, demonstrating improved optimisation and generalisation.
Training, validation, and test accuracies were 0.8270, 0.8001, and 0.7596 respectively, with corresponding loss values of 0.4918, 0.5717, and 0.5996.

The smaller performance gap between training and validation results indicates reduced overfitting, reflecting the effectiveness of batch normalisation, dropout, and extended training epochs.
Early stopping at approximately epoch 20 ensured that the best-performing model weights were preserved while preventing excessive training.

23. MODEL 1VS MODEL 2 COMPARISON

Comparative analysis between Model 1 and Model 2 highlights the impact of architectural and training enhancements on performance.
Model 1 achieved higher training accuracy but exhibited noticeable overfitting, reflected in diverging validation loss.
Model 2, on the other hand, showed improved generalisation with validation and test accuracies of 0.8001 and 0.7596 respectively, and significantly lower validation loss (0.5717). The integration of batch normalisation, dropout layers, and adaptive learning rate scheduling effectively stabilised learning and reduced variance between training and validation performance.
Overall, Model 2 outperformed the baseline in terms of generalisation, robustness, and convergence efficiency.

24. REFLECTION AND CONCLUSION

This project deepened my understanding of convolutional neural networks and their practical implementation for image classification.
I gained valuable insight into how architectural and optimisation techniques such as batch normalisation, dropout, and adaptive learning rates influence model behaviour.
The comparison between the baseline and enhanced CNNs demonstrated how thoughtful model refinement leads to better generalisation and stability.
While the baseline model showed strong initial performance, the enhanced CNN achieved superior validation and test accuracy, confirming its robustness.
Overall, this experience strengthened both my technical and analytical skills in evaluating model performance, while highlighting opportunities for further improvement through data augmentation and transfer learning.