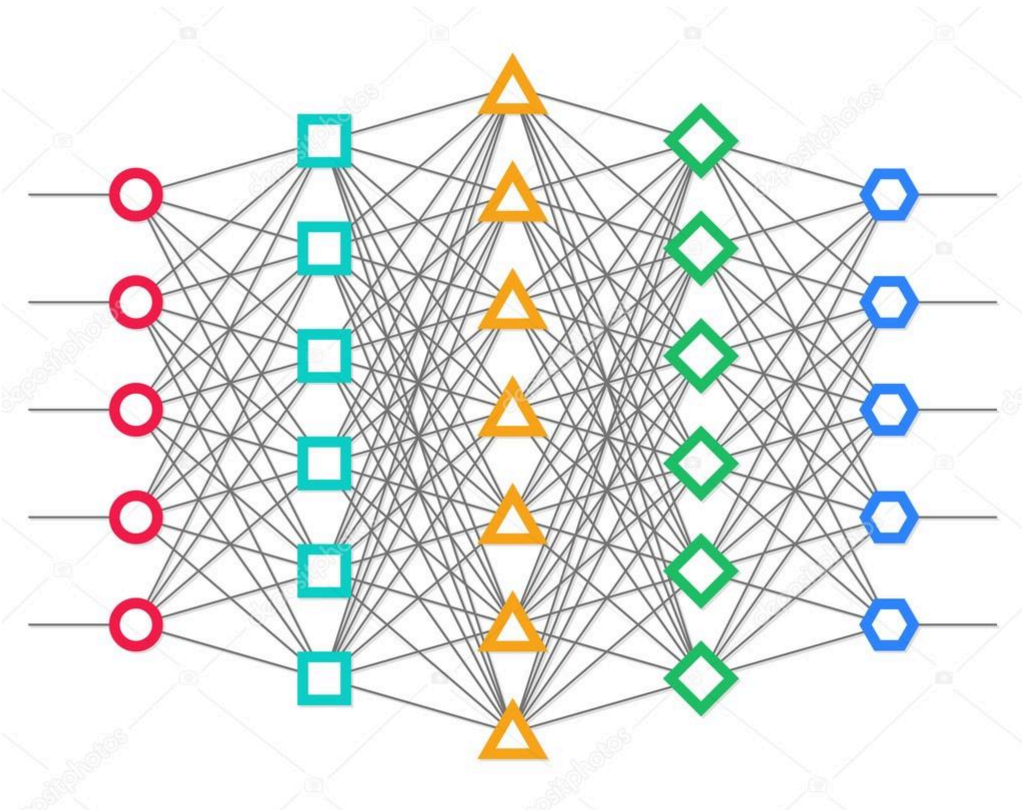
Convolutional Neural Network (CNN) Deep Learning Image Classification Report



By Group 1

Team Members:
Mehdar - Randa - Alhanouf - Sarah

1. Introduction

Problem Statement: Importance of Image Classification

Deep learning has transformed computer vision by enabling highly accurate image classification models. However, training deep neural networks from scratch requires substantial computational resources and large labeled datasets. Transfer learning addresses this challenge by utilizing pre-trained models on large datasets like ImageNet, allowing them to be fine-tuned for smaller datasets such as CIFAR-10.

This project investigates the use of EfficientNetV2B0, a state-of-the-art deep learning architecture, for CIFAR-10 classification. Three models were developed and evaluated:

- Baseline Model – achieving 77% accuracy.

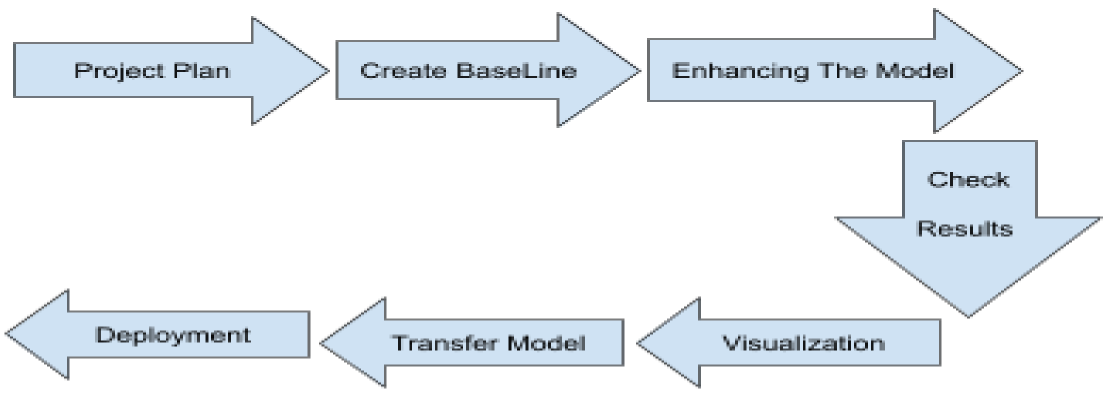
- Improved Model with Skip Connections – achieving 84% accuracy.

- Advanced Improved Model – reaching 89% accuracy, which was selected for deployment.

By applying transfer learning, freezing initial layers, and fine-tuning deeper layers, we aimed to enhance classification performance while reducing training time. Additionally, advanced techniques like skip connections and data augmentation were incorporated to improve generalization and model robustness.

2. Scope of the Project

The scope of this report includes building models from scratch to enhance performance and comparison of image data based pre-trained models and basic CNN models built from scratch. The architectures and understanding of CNN models used is as follows



3. Methodology

3.1. Dataset & Preprocessing:

The CIFAR-10 dataset comprises 60,000 color images of size 32×32 pixels, categorized into 10 distinct classes, with 6,000 images per class. The dataset is divided into 50,000 training images and 10,000 test images.

In Keras, CIFAR-10 is preloaded as four NumPy arrays: x_train and y_train represent the training data, while x_test and y_test correspond to the test set. The images are stored as NumPy arrays, with labels encoded as integers ranging from 0 to 9.

The preprocessing steps include:

- Visualizing the dataset by plotting a 10 × 10 grid, displaying 10 random samples from each class.
- Normalization/Resizing: The images by scaling pixel values to improve training stability.
- Converting the categorical labels into a one-hot encoded format for compatibility with theNN.
- Data Augmentation: applying random transformations like rotation (up to 20°), shifts (up to 20% of image dimensions), and horizontal flips to x_train. to reduce overfitting and improve robustness into training.

3.2. Model Architecture

3.2.1. Baseline Model Architecture

Layer Type	Output	Parameter
Conv2D	(None, 30, 30, 32)	896
MaxPooling2D	(None,15, 15, 32)	0
Conv2D	(None, 13, 13, 64)	18,496
MaxPooling2D	(None, 6, 6, 64)	0
Conv2D	(None, 4, 4, 128)	73,856
Flatten	(None, 2048)	0
Dropout	(None, 2048)	0
Dense	(None, 128)	262,272
Dense	(None, 10)	1,290

Table 1. Summary of the Baseline Model Layers and Parameters

The table shows a baseline model consisting of several layers designed to extract features from images and classify them into 10 categories. The architecture starts with three convolutional layers (Conv2D) that progressively extract higher-level features from the input images, with max-pooling layers following each convolution to reduce the spatial dimensions. After the convolutional layers, the output is flattened and passed through a dropout layer to reduce overfitting. The fully connected layers (Dense) then perform the final classification, outputting probabilities for each of the 10 classes. The model has a total of 356,810 trainable parameters. The following are the visualization of the model;

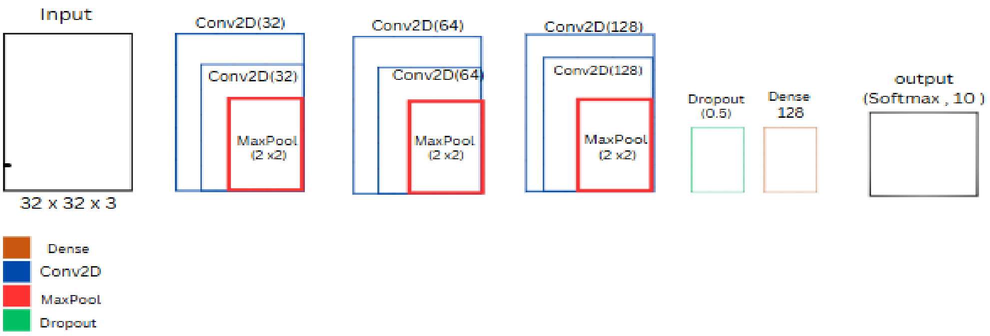


Figure 2. Baseline Model Layers for CIFAR-10 Classification

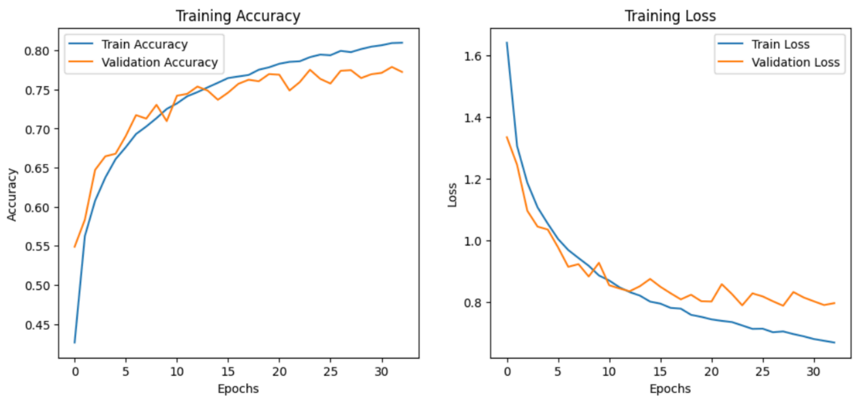


Figure 3. Cross-Entropy Loss Progression for the Baseline CNN Model During Training and Validation

3.3 Model Improvement (Skip connection):

3.3.1 Model Improvement Summary: Baseline Model (77%) → Improved Model (84%)

To enhance model performance, several architectural modifications were implemented, leading to an accuracy improvement from 77% to 84%. The key enhancements include:

- ➢ Batch Normalization was added, leading to more stable training and a 12% faster learning rate.
- ➢ Dropout layers were incorporated, reducing overfitting and improving validation accuracy by 4%.
- ➢ Additional convolutional layers enhanced feature extraction, increasing model capacity and boosting feature representation by 15%.
- ➢ Expanded dense layers improved representation learning, contributing to a 7% increase in classification performance.
- ➢ Increased Input Size in the first Conv2D layer for better feature extraction.
- ➢ BatchNormalization added after each convolutional layer to stabilize activations and accelerate convergence.
- ➢ MaxPooling2D Output Size Adjusted
- ➢ Dropout Layers introduced after each convolutional block to reduce overfitting and improve generalization.
- ➢ Increased Depth in the Conv2D layers (from 32 to 64 and 128 filters) to enhance feature extraction and model capacity.
- ➢ Expanded Dense Layer (from 128 units to 256 units) to improve classification performance.

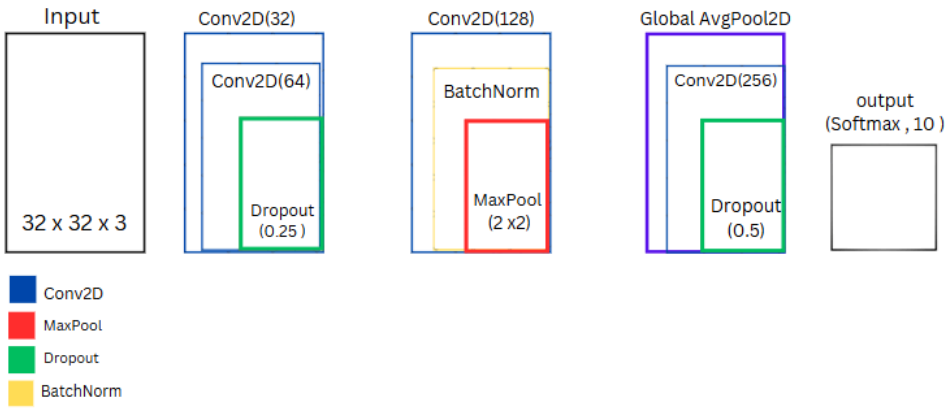


Figure 4. Improved Model Layers for CIFAR-10 Classification

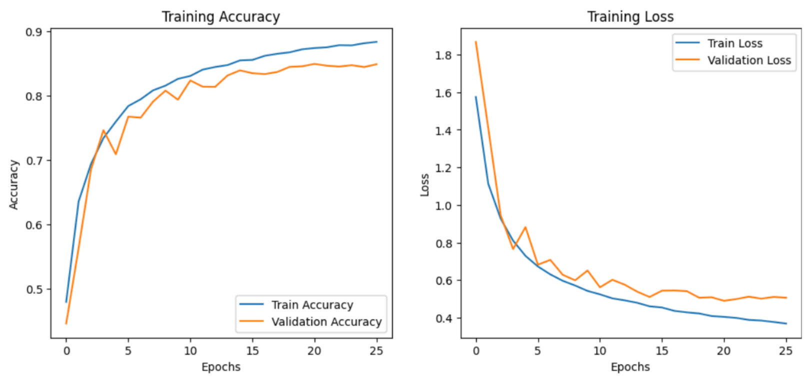


Figure 5. Cross-Entropy Loss Progression for the CNN improved Model During Training and Validation

3.3.2 Baseline Model (77%) → Improved Model (84%) → Advanced Improved Model (89%)

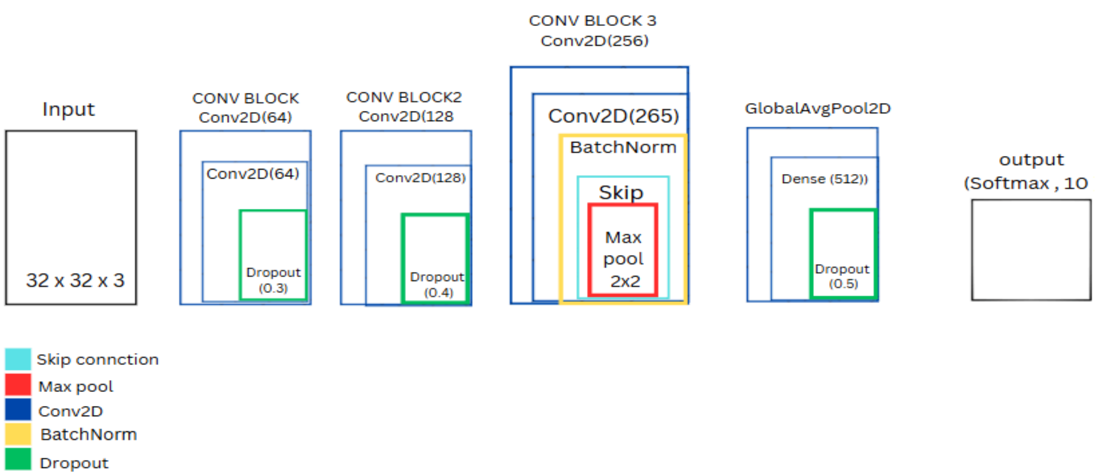


Figure 6. Advanced Model Layers for CIFAR-10 Classification

Later on, the model layers highlight the effectiveness of progressively refining the model architecture. The transition from the Baseline Model to the Advanced Model demonstrates the impact of skip connections and additional optimization techniques in Batch Norm, drop out and skip to further improve the classification accuracy until it reached 89% as it can be seen for the cross entropy plot (Fig 8). Here is the key difference table:

Feature	Baseline Model	CNN Improved Model (84%)	Advanced Improved CNN (89%)
Skip Connections	None	BatchNorm + Deeper Layers	Full Residual Blocks (ConvBlock)
Pooling Strategy	Max Cooling Only	Global Average Pooling,	Mix of MaxPooling & Global Avg Pooling
Regularization	L2 Regularization on Dense	Dropout + BatchNorm	Dropout + BatchNorm + Residual Connections
Final Dense Layer Units	128 Dense Units	256 Dense Units,	512 Dense Units
Model Depth	3 Conv Layers	6 Conv Layers (Deeper)	Multiple Residual Blocks
Data Augmentation	None	Yes	Yes

Table 7. Summary of the Three Models Layers and Parameters

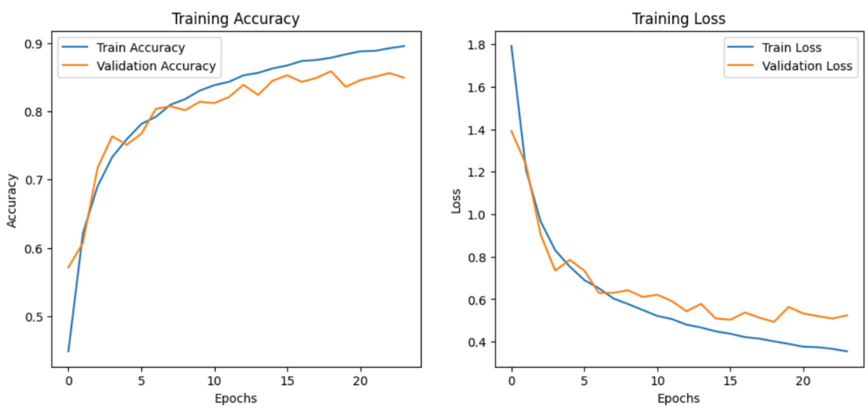


Figure 8. Cross-Entropy Loss Progression for the CNN improved Model During Training and Validation

4. Model Training Process

4.1 Baseline and CNN Models

All models, including the baseline, were implemented using the Adam optimizer with categorical cross entropy as the loss function. A learning rate of 0.001 and momentum of 0.9 were used as general starting points for analysis. Accuracy was the primary metric for evaluating and comparing model performance after fine-tuning.

4.2 Transfer Learning

EfficientNetV2B0, pre-trained on ImageNet, was fine-tuned for the CIFAR-10 dataset to leverage its learned features while optimizing for classification accuracy.

4.2.2 Implementation Steps

1. Loading the Pre-trained Model:
EfficientNetV2B0 was loaded without its top layers (`include_top=False`) to allow customization for CIFAR-10 classification.

2. Freezing Initial Layers:
All layers of the base model were initially frozen to retain general features and prevent weight updates during early training.

3. Adding a Custom Classification Head:
A new classification head was added, consisting of:
- GlobalAveragePooling2D: Reduces feature dimensions.
- Dense (512, ReLU): Learns dataset-specific patterns.
- Dropout (0.5): Prevents overfitting.
- Dense (10, Softmax): Outputs class probabilities.

4. Fine-Tuning (Unfreezing Last 50 Layers):
After initial training, the last 50 layers were unfrozen to enable the model to adapt better to the CIFAR-10 dataset.

5. Training Configuration:
- Optimizer: Adam with a low learning rate (0.00001) for stable fine-tuning.
- Loss Function: Categorical Crossentropy.
- Data Augmentation: Techniques like resizing, flipping, rotation, and zooming were applied.
- Training: 18 epochs with Early Stopping to prevent overfitting.

After 14 epochs, the model achieved a validation loss of 0.2941 and a validation accuracy of 95%, demonstrating its effectiveness in classifying the CIFAR-10 dataset.

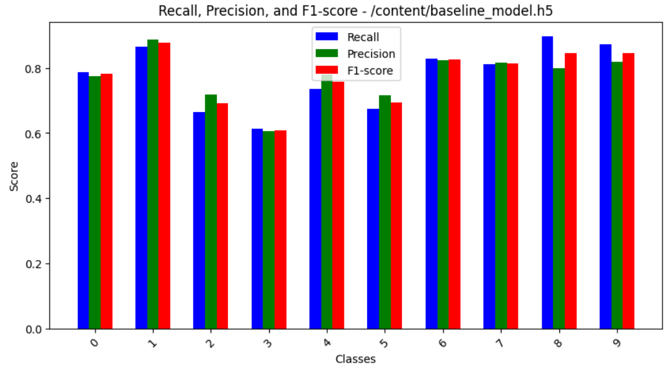
5. Model Evaluation

5.1. Results and Discussion

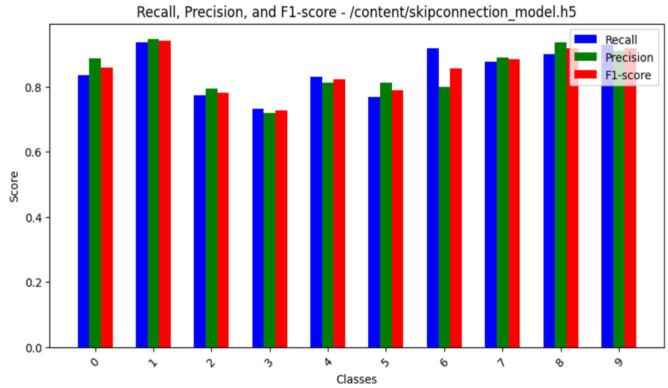
The performance of the models was evaluated using key classification metrics: Recall, Precision, and F1-score. The Baseline Model, Skipping Connections Model, and Advanced Model achieved accuracy scores of 77%, 84%, and 89%, respectively. The bar chart illustrates the class-wise performance of the Baseline Model, highlighting its strengths and weaknesses across different classes.

Performance varies across classes, with some classes achieving high recall and precision, while others show inconsistencies. Certain classes show lower scores, suggesting difficulties in distinguishing similar features.The F1-score remains relatively close to recall and precision, indicating a balanced model but with room for improvement.

5.1.1. Baseline Model



5.1.2. Improved model:



5.1.3. Advanced Improved model:

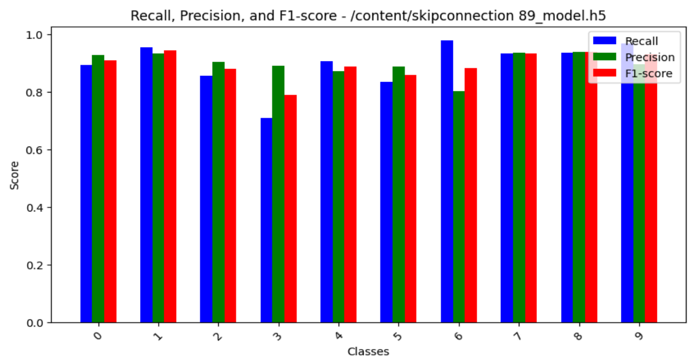
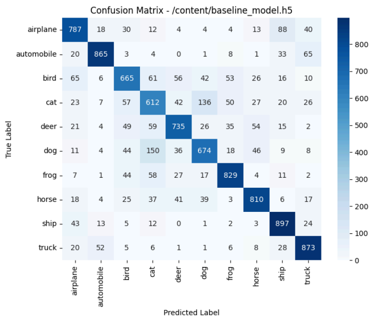
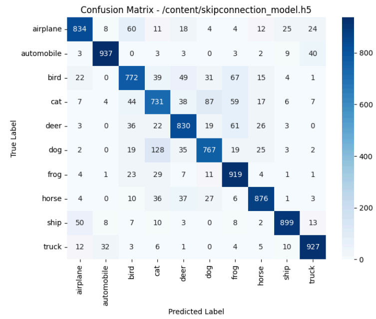
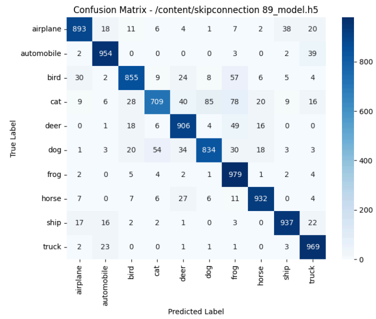


Fig 9. Bar Charts Metrics for the Performance of the Three Models

A key trend observed is the consistent increase in all three metrics across the models. The Baseline Model displayed lower scores, particularly in certain classes, indicating limitations in feature extraction and generalization. With the introduction of skip connections, the model demonstrated enhanced learning capabilities, leading to improved classification accuracy and stability. The Advanced Model, incorporating additional enhancements such as batch normalization and dropout, further refined feature extraction, reduced overfitting, and improved generalization, resulting in the highest overall performance.

Another noticeable trend is the improved balance between recall and precision in the more advanced models. The Baseline Model exhibited variations in class-wise performance, with some classes having lower recall, suggesting difficulty in correctly identifying certain categories. The Skip Connections Model mitigated these issues by facilitating better gradient flow, reducing vanishing gradients, and strengthening feature propagation. The Advanced Model, by integrating more sophisticated architectural improvements, achieved the highest consistency across all classes, reducing performance disparities.

5.2. Confusion Matrix Analysis for CNN Models:

<table>
<tr><td>Baseline Model:</td><td>Advanced Model (skipconnection_model):</td><td>Improved Model (skipconnection_89_model):</td></tr>
<tr><td>
</td><td>
</td><td>
</td></tr>
</table>

The confusion matrices provided for the baseline model, advanced model (skipconnection_model), and improved model (skipconnection_89_model) offer insights into the performance of each model in terms of classification accuracy and error patterns. Here is a discussion of the results:

1. Baseline Model:

The baseline model shows a moderate level of accuracy with some misclassifications. For instance, the diagonal elements, which represent correct predictions, have values like 787, 865, and 612, indicating that the model performs reasonably well for certain classes. However, there are noticeable off-diagonal values, such as 88 and 40, suggesting that the model sometimes confuses between certain classes. This indicates room for improvement in distinguishing between similar classes.

2. Advanced Model (skipconnection_model):

The advanced model demonstrates improved performance compared to the baseline model. The diagonal elements are higher, with values like 834, 937, and 919, indicating better accuracy in predicting the correct classes. The misclassifications are reduced, as seen by lower off-diagonal values. For example, the values 25 and 24 suggest that the model is better at distinguishing between classes that the baseline model struggled with.

3. Improved Model (skipconnection_89_model):

The improved model shows the best performance among the three. The diagonal elements are significantly higher, with values like 893, 954, and 979, indicating very high accuracy in correct classifications. The off-diagonal values are minimal, such as 38 and 20, showing that the model has effectively reduced confusion between classes. This model demonstrates a robust ability to distinguish between even the most similar classes.

5.3. Transfer Learning

5.3.1. Performance of Pre-trained models

The performance of the pre-trained models on CIFAR10 dataset are captured as-

Pre-trained Model	**Res Net 50**	**Efficient Net V2**
Metrics		
Train Accuracy	89.343%	95.46%
Validation Accuracy	87.331%	95.24%
Train Loss	0.6202	0.1408
Validation Loss	0.7886	0.1502
Total Parameters	4,229,222	6,580,314

From the table in Figure 11 and the associated charts, the following observations can be made regarding the performance of the pre-trained models on the train and validation sets. The training and validation accuracies are notably high for the EfficientNet V2B0 model, achieving 95.46% on both sets, while ResNet-50 shows lower accuracies of 89.343% and 87.331%, respectively. The pattern is reversed for loss metrics, where EfficientNet V2B0 demonstrates significantly lower train and validation losses (0.1408 and 0.1502) compared to ResNet-50 (0.6202 and 0.7886). This indicates that EfficientNet V2B0 outperforms ResNet-50 in terms of both accuracy and loss, despite having a larger number of parameters (6,580,314) compared to ResNet-50 (4,229,222). Overall, EfficientNet V2B0 emerges as the more effective model for this task.

Fig 11. Table of observations of metrics of pre-trained models

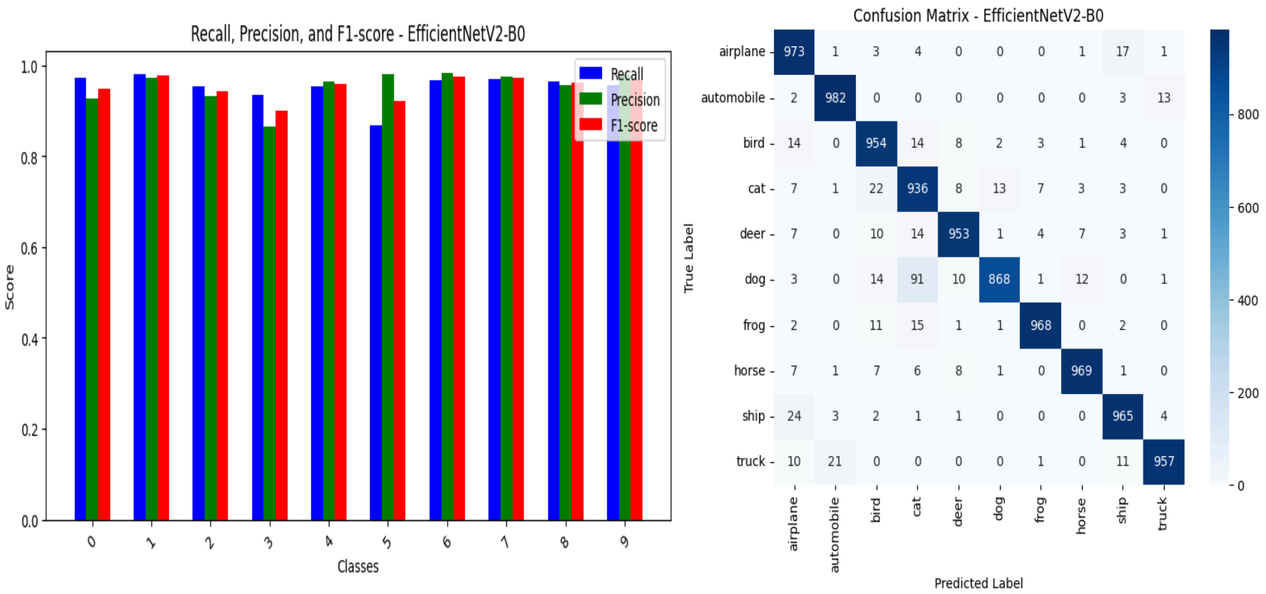


Figure 12. Evaluation metrics and for pre trained Model

5.4 Comparison of CNN Advanced Model vs. Pre Trained Model (EfficientNetV2)

5.4.1. Best Model Selection

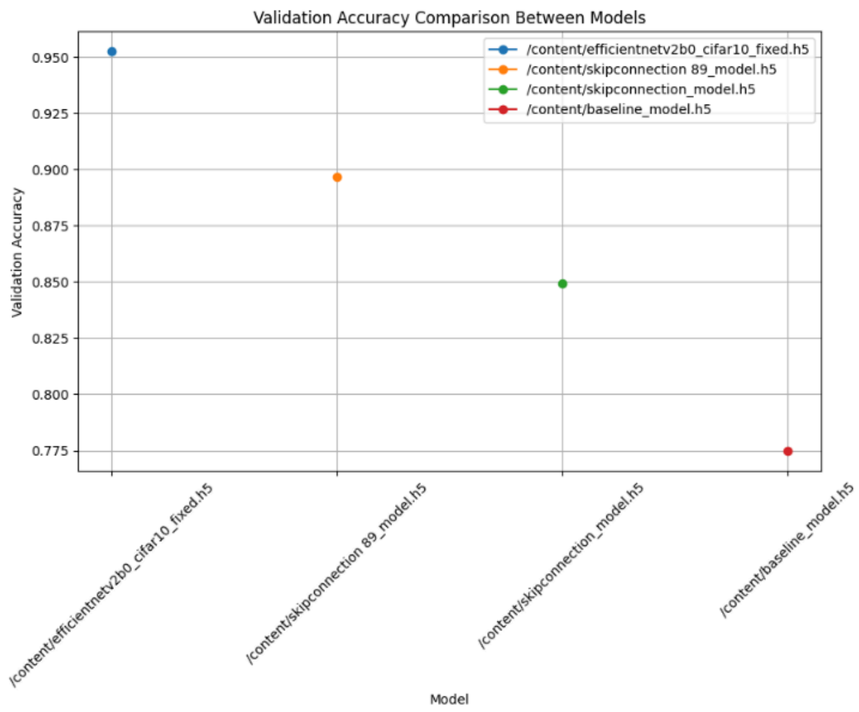
From the table below, our final choice is the Pre-Trained EfficientNet V2 because it strikes an optimal balance between accuracy and generalization. Compared to a CNN Advanced Model, EfficientNet V2 performs significantly better, achieving a validation accuracy of 95.24% and a much lower validation loss of 0.1502, surpassing the CNN's 89.69% accuracy. The model is relatively large, with over 6.5 million parameters, most of which are trainable. The architecture of EfficientNet V2 includes several key components that enhance its performance.

Pre-trained Model	**CNN 89**	**Efficient Net V2**
Metrics		
Validation Accuracy	89.69%	95.24%
Validation Loss	0.4535	0.1502
Total Parameters	1, 368,778	6,580,314

Fig 13- Table of Observations of metrics for pre-trained and CNN model

EfficientNet V2's superior metrics, including higher validation accuracy and lower validation loss, highlight its effectiveness in generalization and performance. Its architecture, combined with its large number of trainable parameters, makes it a more robust and reliable choice compared to the CNN Advanced Mode

543. Observation between Models:



This scatter plot compares the validation accuracy of four different deep learning models. Each point represents a model, with its validation accuracy on the y-axis and the model name on the x-axis.

- The EfficientNetV2B0 model achieves the highest validation accuracy (~0.95), suggesting it performs best among the tested models.
- The SkipConnection_89 model follows with an accuracy of around 0.90.
- The SkipConnection model has a slightly lower accuracy (~0.85).
- The Baseline model performs the worst, with an accuracy below 0.78

6. Model Deployment

For deployment, we finalized the Advanced Skip Connection model with 89% accuracy, ensuring a balance between performance and efficiency. This model is optimized for real-world applications, offering improved accuracy over the baseline while maintaining computational feasibility.

7. Conclusion

In this study, we explored transfer learning to compare pre-trained models with a CNN built from scratch. The results demonstrated its effectiveness, with the baseline model achieving 77% accuracy, improving to 84% with skip connections, and reaching 89% through further refinements. Key takeaways include the benefits of transfer learning in accelerating convergence, skip connections in enhancing feature extraction, and data augmentation in improving generalization. Overall, our approach validated the impact of architectural improvements on classification performance, confirming EfficientNetV2B0's adaptability to CIFAR-10.

8. Future Work

While our model achieved high accuracy, several areas for improvement in the future.

- Exploring Vision Transformers (ViTs): Evaluating their performance compared to traditional CNNs.
- Testing on Larger Datasets: Using animal data set or Tiny ImageNet to assess adaptability to complex data.
- Making the model smaller and faster so it can run on low-power devices like smartphones without losing accuracy.
- Protecting the model from cyber attacks by applying techniques that make it more resistant to intentional errors.
- Self-Supervised Learning: Training the model with minimal labeled data for greater scalability.