DEEP LEARNING

Frank Sarfo
Lokesh
Rafael Dias Rocha

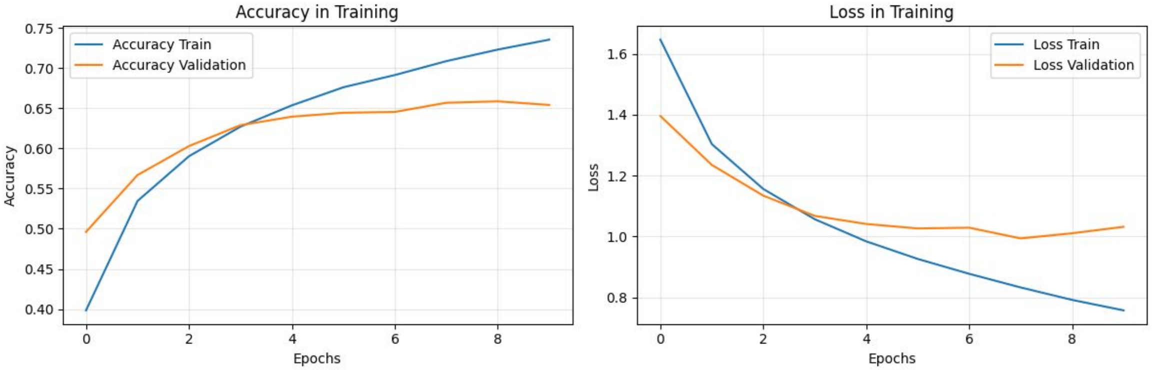
Basic CNN - Architecture & Performance

Initial Approach & Findings:

We developed a custom CNN with three convolutional layers of increasing filter complexity (32→64→64).

The model achieved strong training accuracy (78.2%), but showed clear signs of overfitting, with validation performance dropping to 68.6%.

This 9.6% generalization gap indicates the model was memorizing training data patterns rather than learning robust, transferable features.



CNN WITH DATA AUGMENTATION

Enhanced Methodology & Results:

To combat overfitting, we implemented a comprehensive regularization strategy including data augmentation (rotations, shifts, flips), batch normalization, and progressive dropout (25-50%).

This approach yielded remarkable improvements: test accuracy increased to 76.8% (+8.2% improvement).

The model demonstrated significantly better generalization across all classes.

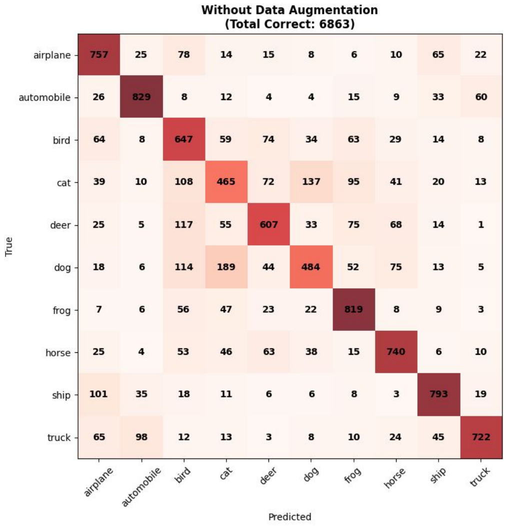
COMPARISON: WITH vs WITHOUT AUGMENTATION

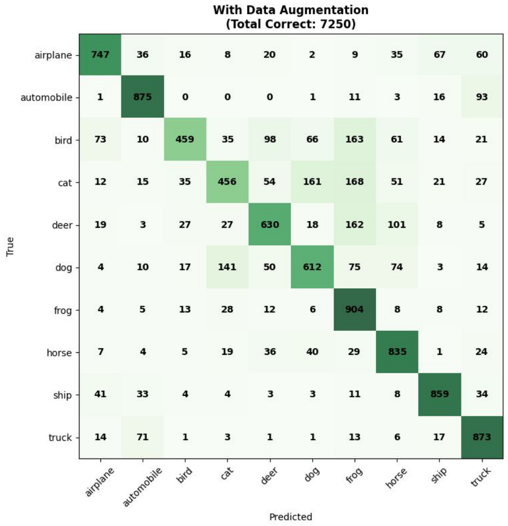
Comparative Performance Analysis:

Our systematic evaluation revealed substantial improvements across all metrics.

The CNN with data augmentation achieved 76.8% test accuracy compared to 68.6% without augmentation - an 8.2% improvement.

This significant gain demonstrates data data augmentation effectiveness in enhancing model generalization and reducing overfitting, leading to more robust feature learning.





TRANSFER LEARNING WITH MOBILENETV2

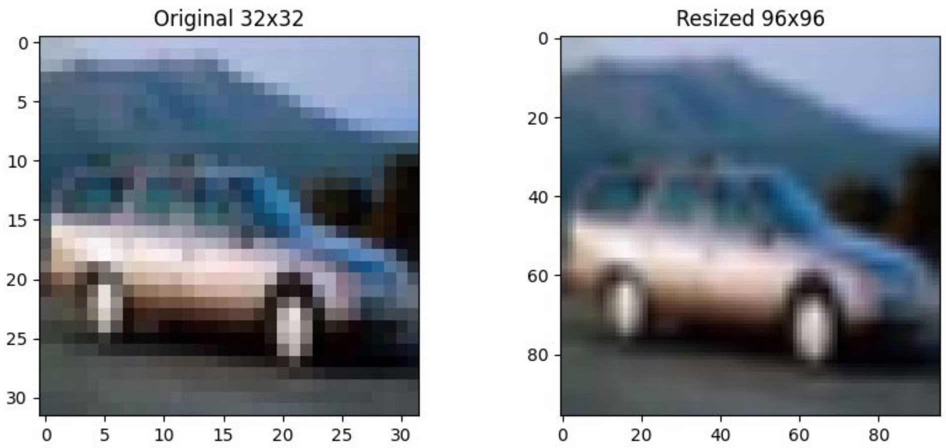
Advanced Strategy Implementation:

We leveraged MobileNetV2 pre-trained on ImageNet, employing a sophisticated two-phase training approach.

A critical preprocessing step involved resizing images from 32×32 to 96×96 pixels, ensuring compatibility with MobileNetV2's architecture while preserving essential visual information.

The base network was initially frozen while custom classification layers were trained, followed by careful fine-tuning of the final 50 layers.

This strategy capitalized on established feature extraction capabilities while adapting the model to our specific CIFAR-10 classification task, significantly reducing training time and computational requirements.

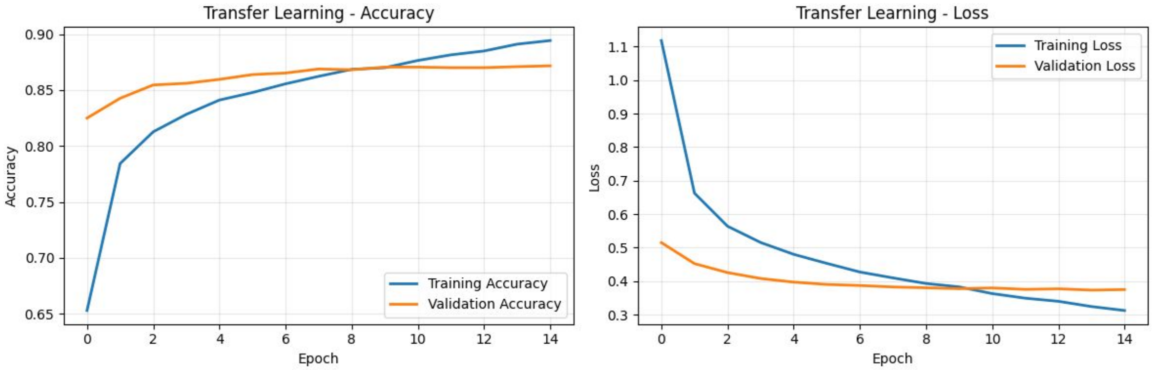


TRANSFER LEARNING RESULTS

Performance Breakthrough:

The transfer learning approach achieved exceptional results with 91.23% test accuracy - a 32.9% improvement over our baseline CNN and 18.8% improvement over the augmented model.

The model demonstrated outstanding performance on distinct categories like ships (95%+) and airplanes (93%+), while maintaining consistent accuracy across all classes with minimal overfitting and stable convergence.



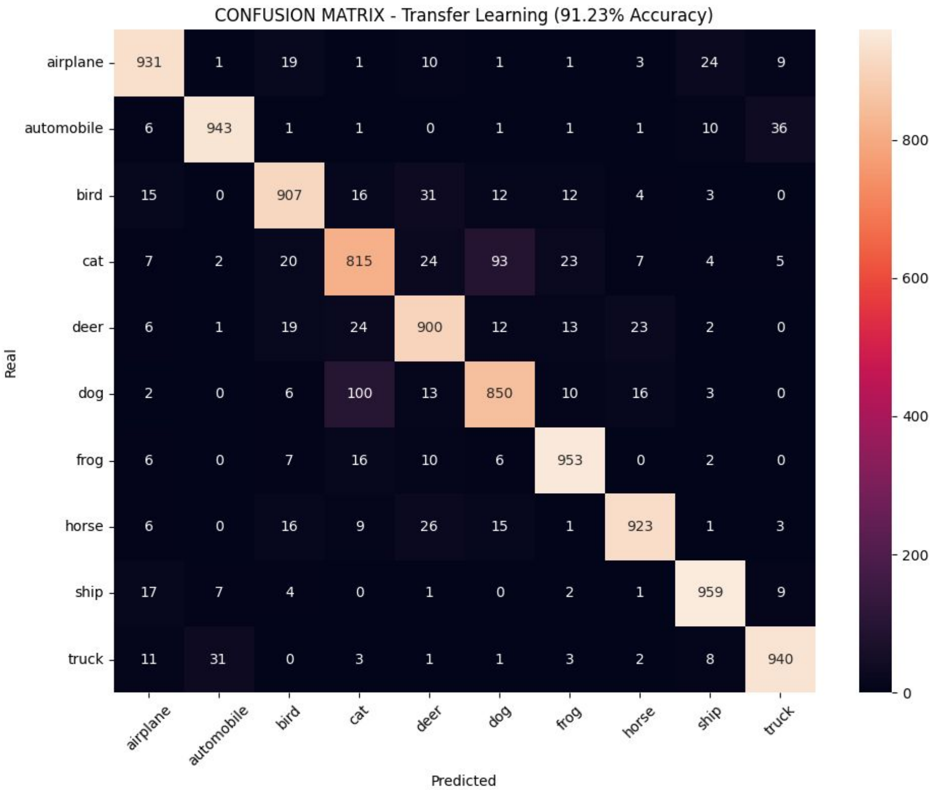
DETAILED TL ANALYSIS

Comprehensive Performance Assessment:

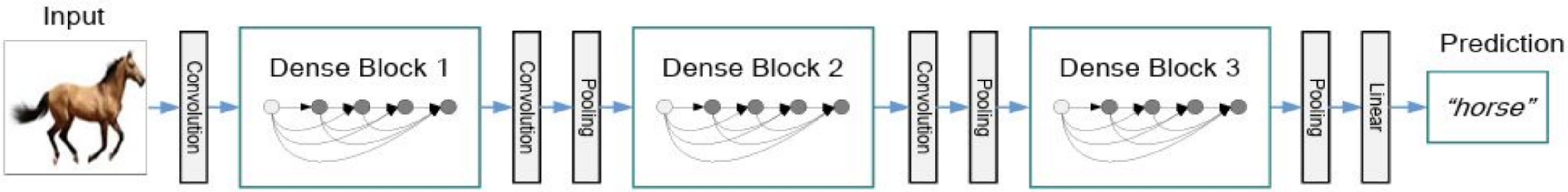
Despite the outstanding 91.23% overall accuracy, our analysis reveals persistent challenges in distinguishing semantically similar categories.

Cat classification remained the most challenging at approximately 86%, with common confusions occurring between cats/dogs and birds/airplanes.

However, transfer learning conclusively proved superior, delivering the optimal balance between performance (91.23%), training efficiency, and generalization capability while leveraging pre-existing feature knowledge.

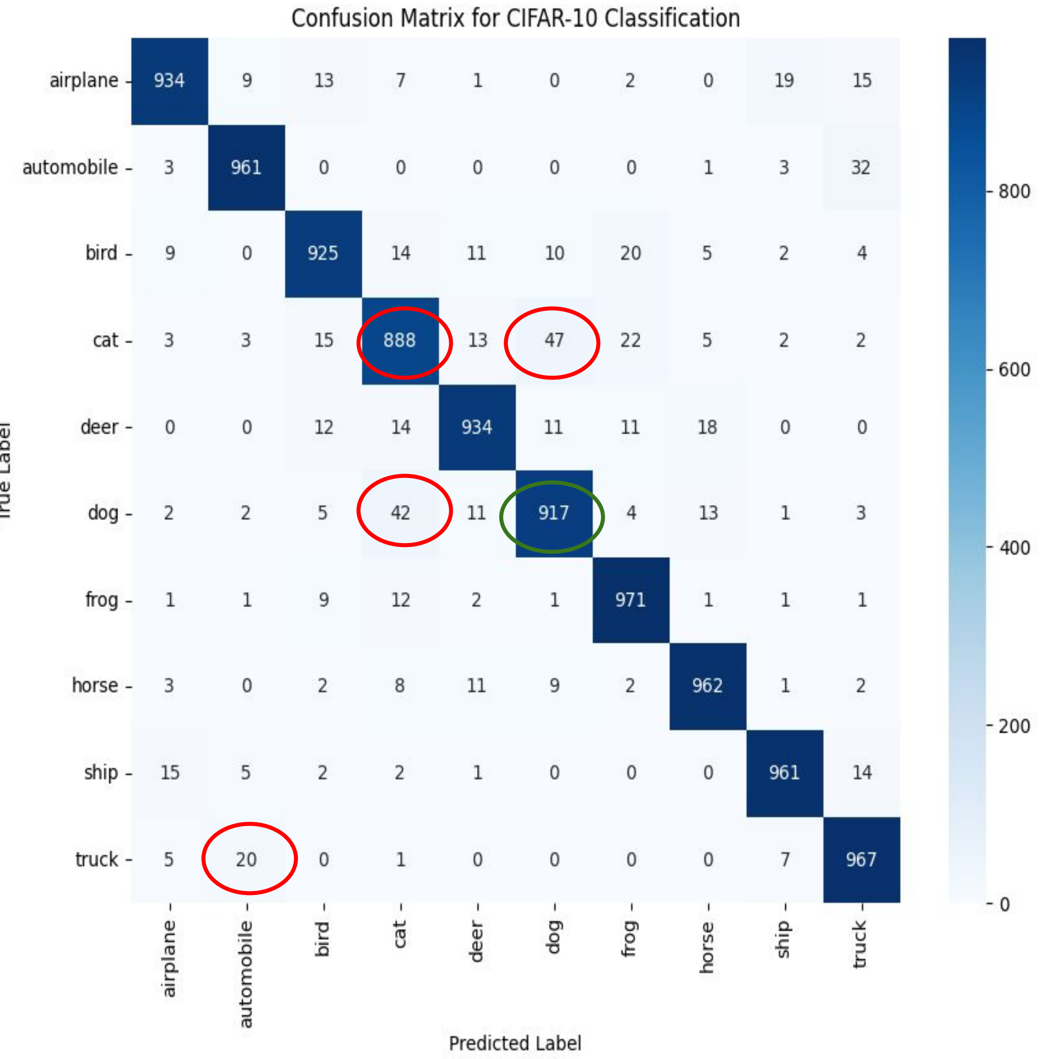


DenseNet 121



- Dense connectivity boosts feature reuse.
- Low parameter count, high efficiency.
- Strong accuracy for image classification tasks.

Model	Model Size(Mb)	Top 1 acc(%)	Top 5 acc(%)	Parameters(M)	Depth
DenseNet121	33	74.91%	92.38%	8.0	121




```
================================================================
CLASSIFICATION REPORT
================================================================
```

	precision	recall	f1-score	support
airplane	0.96	0.93	0.95	1000
automobile	0.96	0.96	0.96	1000
bird	0.94	0.93	0.93	1000
cat	0.90	0.89	0.89	1000
deer	0.95	0.93	0.94	1000
dog	0.92	0.92	0.92	1000
frog	0.94	0.97	0.96	1000
horse	0.96	0.96	0.96	1000
ship	0.96	0.96	0.96	1000
truck	0.93	0.97	0.95	1000
accuracy			0.94	10000
macro avg	0.94	0.94	0.94	10000
weighted avg	0.94	0.94	0.94	10000

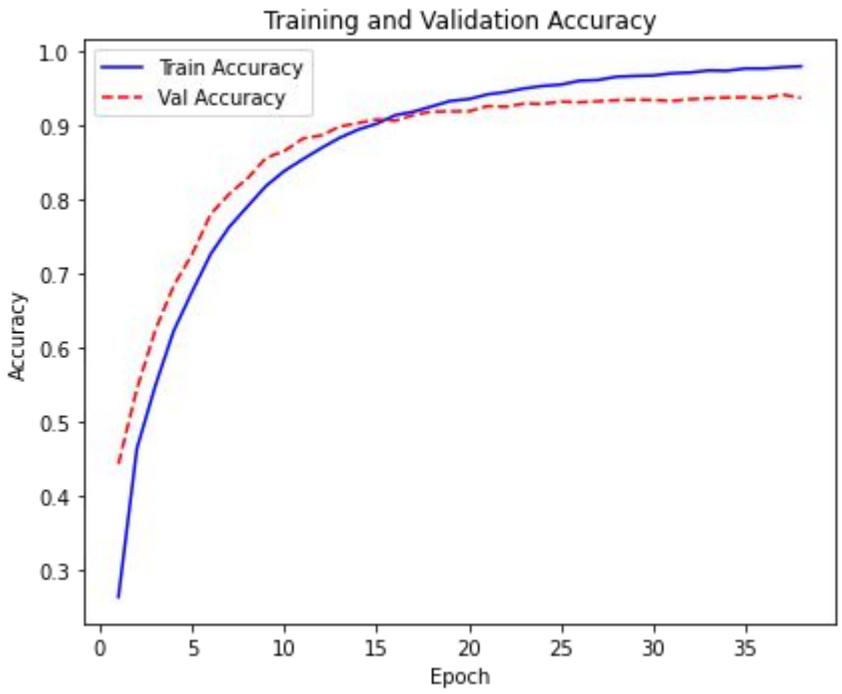
DenseNet 121

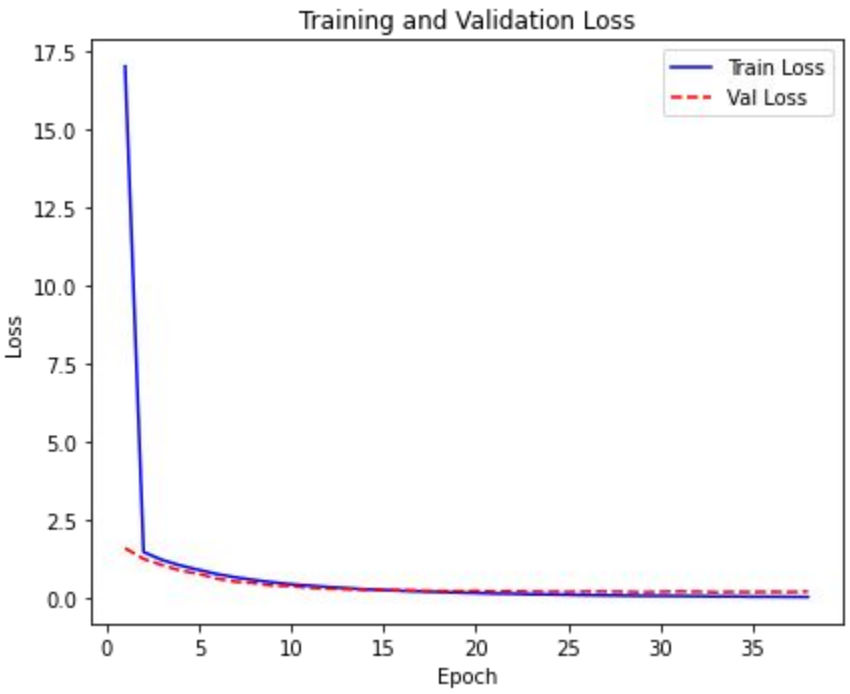
Model	Train acc(%)	Valid acc(%)	Test acc(%)	Train Loss	Valid Loss
DenseNet121 (base Model)	32%	35%		1.81	1.78
DenseNet121 (Fine Tuning)	98%	94.2%	94.2%	0.06	0.21

Fine Tuning:

- **Data augmentation(Rotation, Translation, Zooming, Flipping)**
- **Freezing Layers(First Layers, Last Layers)**
- **Classification head**
- **Callbacks**

Densenet 121 Plots

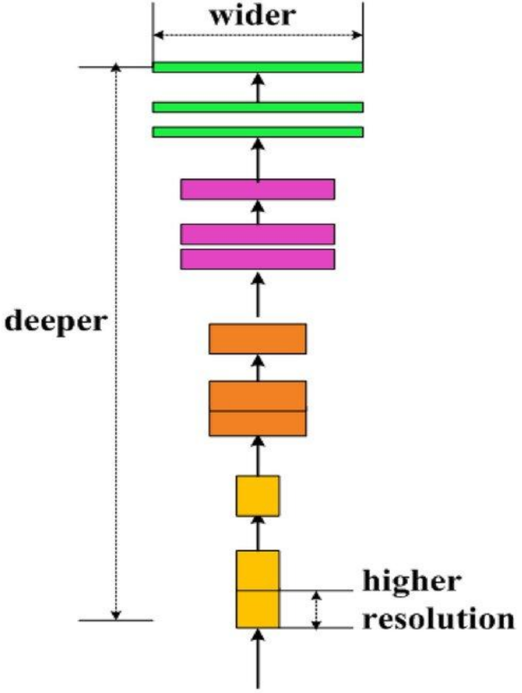




- Accuracy improves rapidly and plateaus above 93%.
- Training and validation loss steadily decrease and stay close.
- No signs of overfitting; model generalizes well to unseen data.

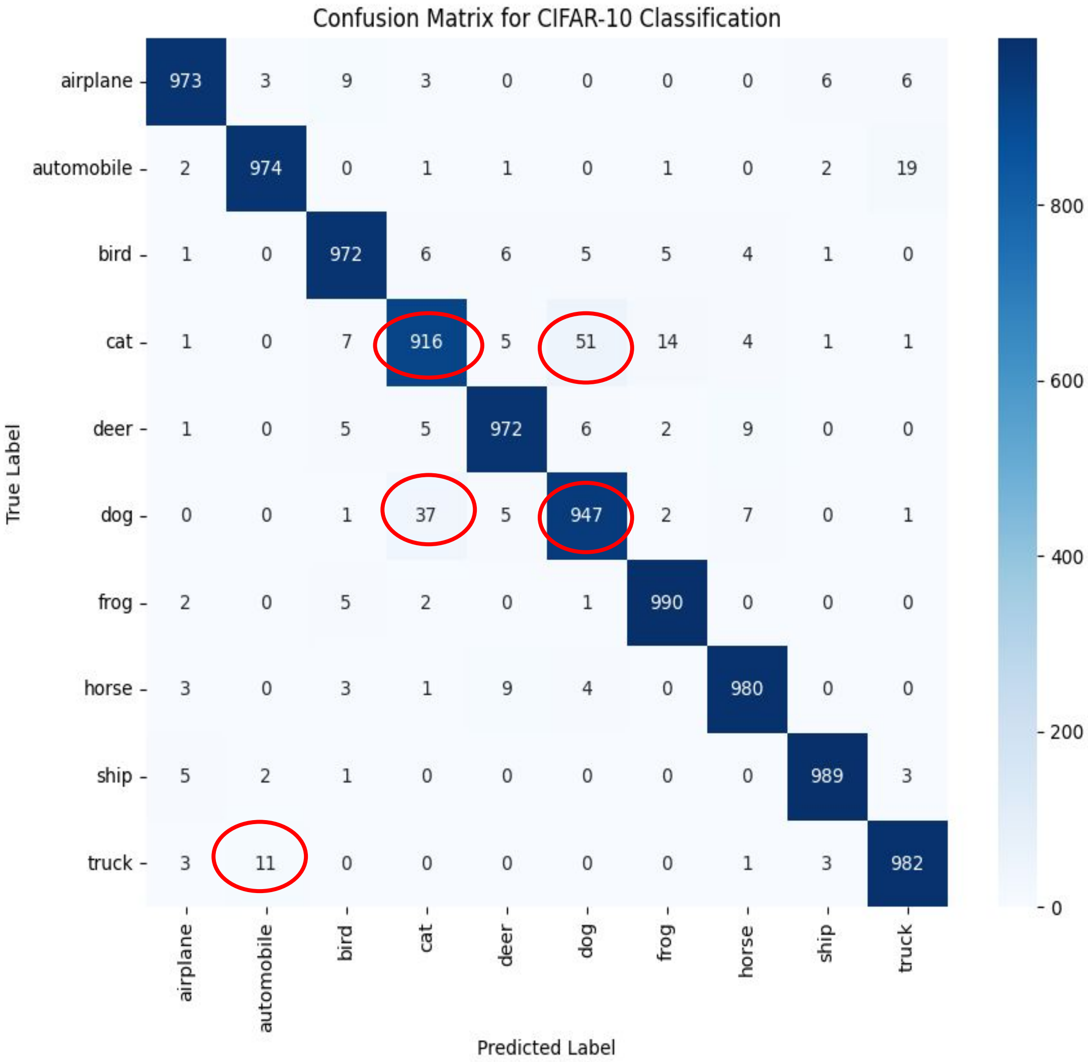
EfficientNet_V2_B0

- High accuracy, low compute.
- Compound scaling: depth, width, resolution.
- Small models, fast inference.



(b) compound scaling

Model	Model Size(Mb)	Top 1 acc(%)	Top 5 acc(%)	Parameters(M)	Depth
EfficientNetV2B0	28	78.7%	94.3%	7.2	



CLASSIFICATION REPORT

	precision	recall	f1-score	support
airplane	0.98	0.97	0.98	1000
automobile	0.98	0.97	0.98	1000
bird	0.97	0.97	0.97	1000
cat	0.94	0.92	0.93	1000
deer	0.97	0.97	0.97	1000
dog	0.93	0.95	0.94	1000
frog	0.98	0.99	0.98	1000
horse	0.98	0.98	0.98	1000
ship	0.99	0.99	0.99	1000
truck	0.97	0.98	0.98	1000
accuracy			0.97	10000
macro avg	0.97	0.97	0.97	10000
weighted avg	0.97	0.97	0.97	10000

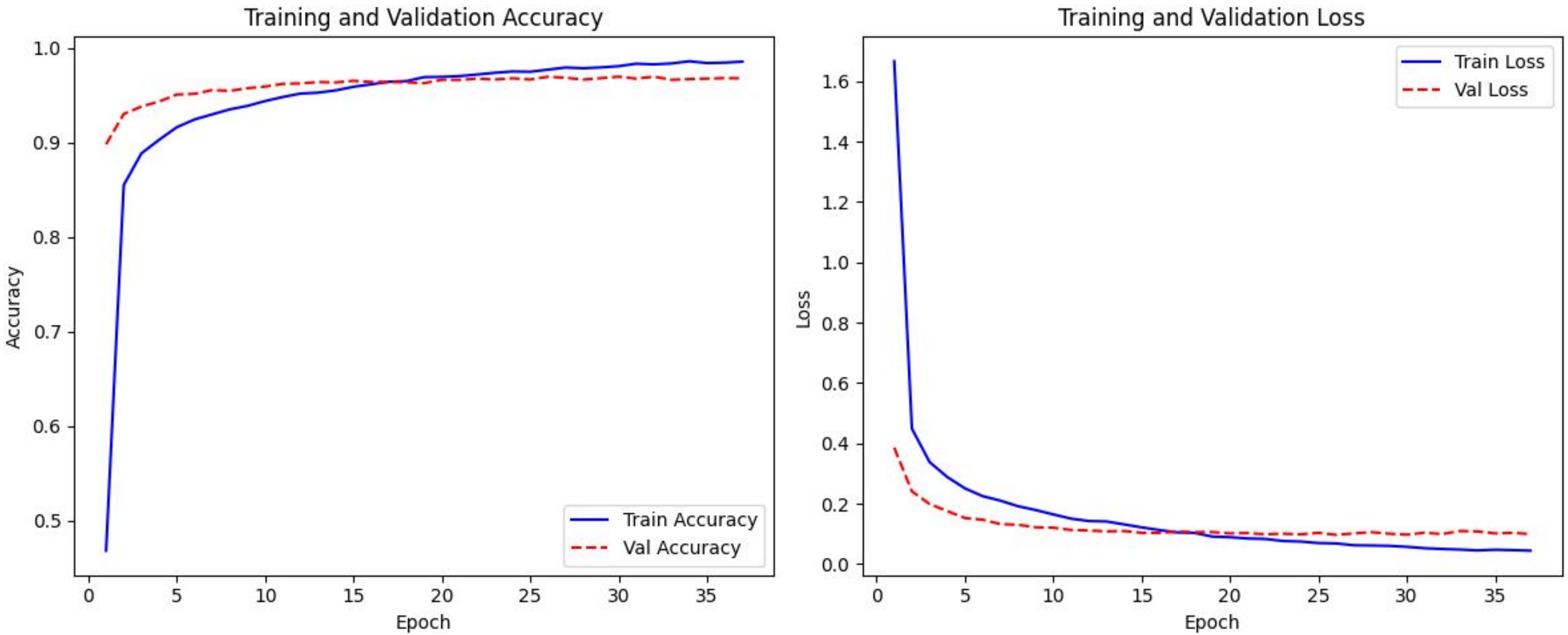
EfficientNetV2_B0 Plots

Model	Train acc(%)	Valid acc(%)	Test acc(%)	Train Loss	Valid Loss
EfficientNetV2_bo (base Model)	48%	42%		1.2	1.46
EfficientNetV2_bo (Fine Tuning)	98%	97%	97%	0.057	0.096

Fine Tuning:

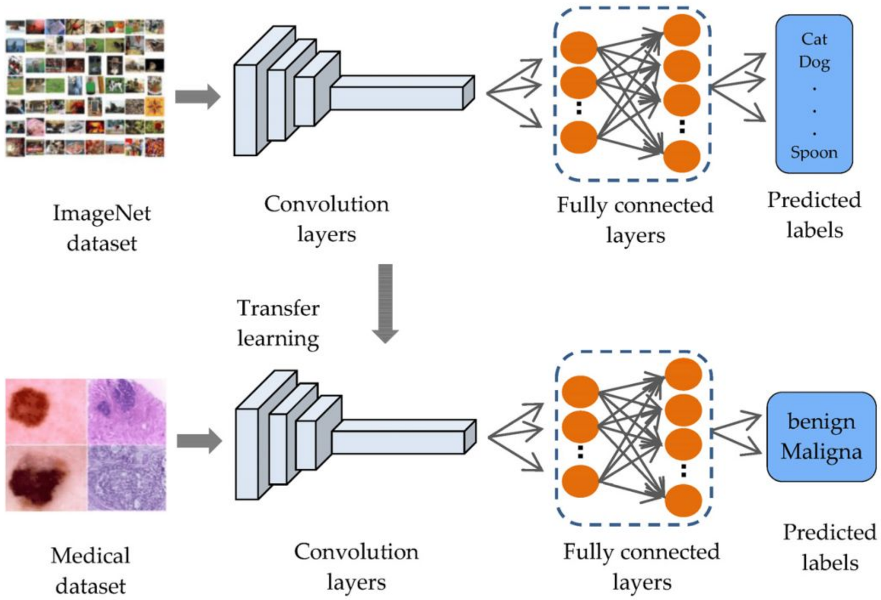
- **Data augmentation(Rotation, Translation, Zooming, Flipping)**
- **Freezing Layers(First Layers, Last Layers)**
- **Classification head**
- **32*32*3—---------> 164*164*3**
- **Callbacks**
- **Learning Rate control**

EfficientNetV2_B0 Plots



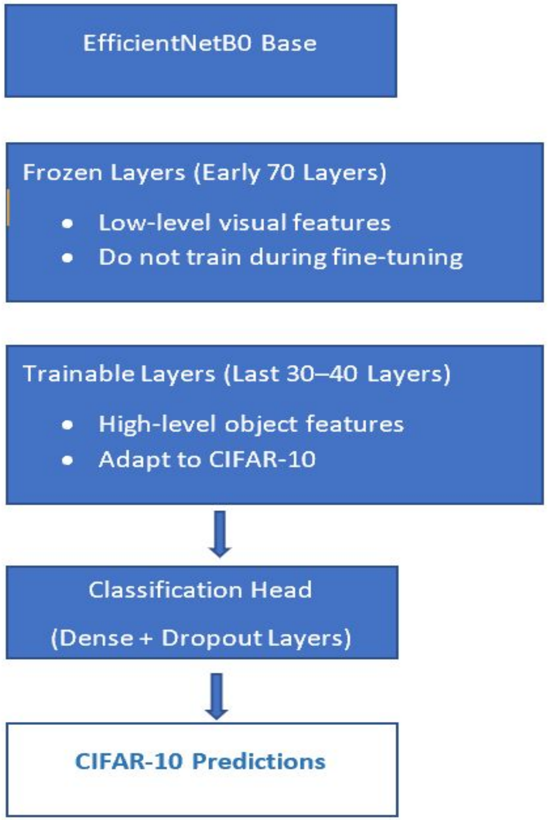
- Accuracy improves rapidly and exceeds 95% for both train and validation sets.
- Loss values decrease and remain closely matched, indicating stable and effective learning.
- No sign of overfitting, as training and validation curves track closely throughout.

EfficientNetB0 Architecture Overview



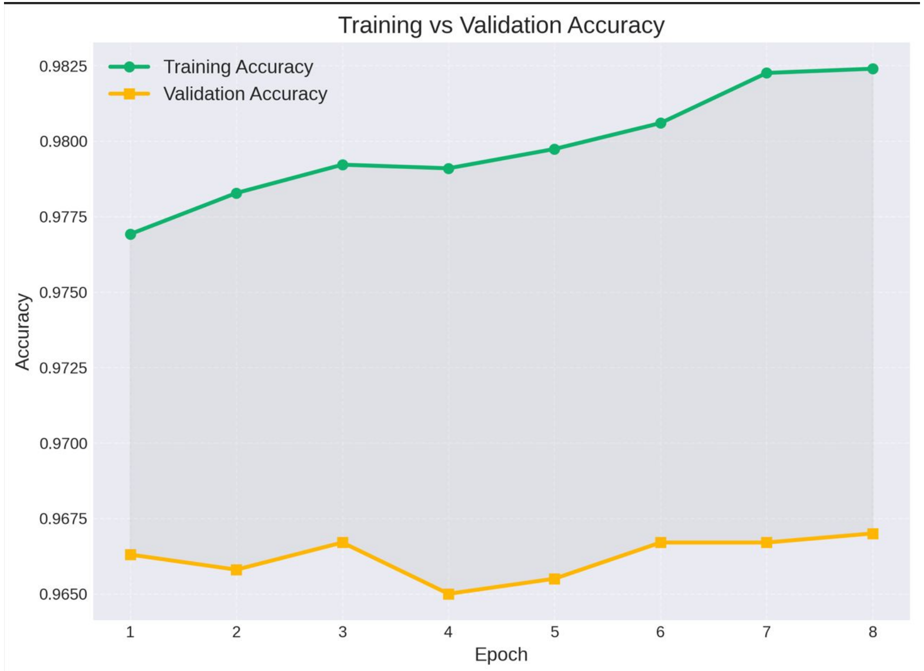
- Pretrained on ImageNet → strong feature extractor
- Convolution layers reused for CIFAR-10 images
- Custom dense layers added for 10-class classification
- Fine-tuned top layers for improved accuracy & generalization

Selective Fine-Tuning Strategy (EfficientNetB0)



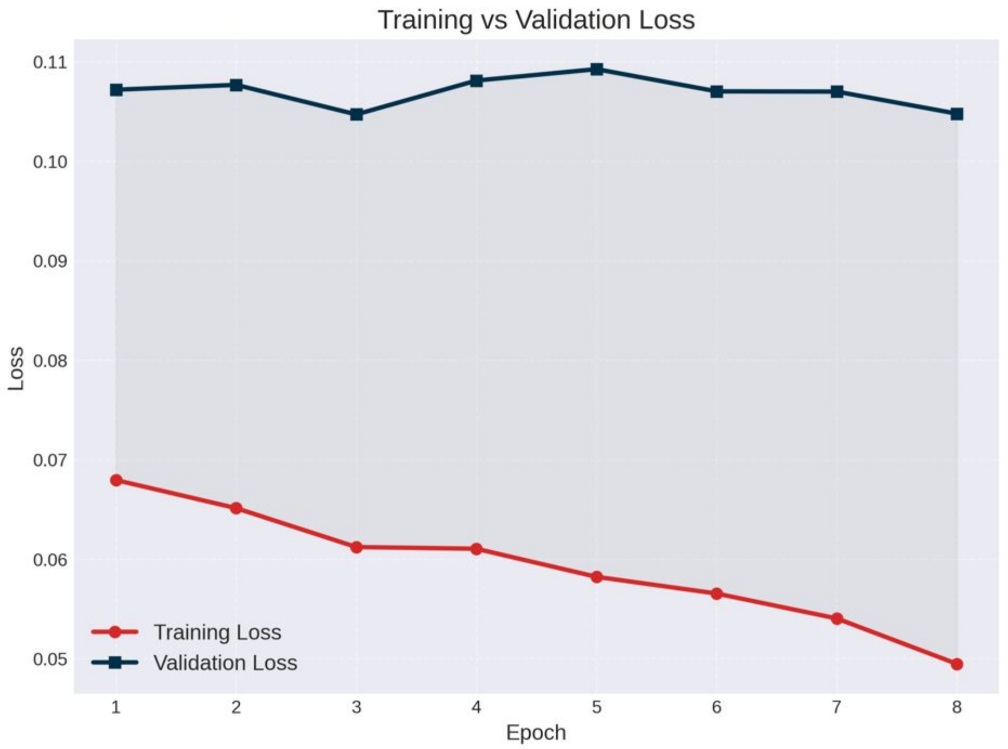
- Phase 1: Train classification head only
- Phase 2: Unfreeze the last 30−40 layers
- Freeze first 70+ layers for stable transfer learning
- Use low learning rate (1e-5) to avoid damaging pretrained weights
- Benefits: Prevents catastrophic forgetting & improves class-specific learning

Training Progress: Accuracy



- The training accuracy curve shows steady improvements
- The validation accuracy curve follows a similar trend
- A small gap between training and validation accuracy indicates minimal overfitting
- Fine-tuning phase gives the biggest jump

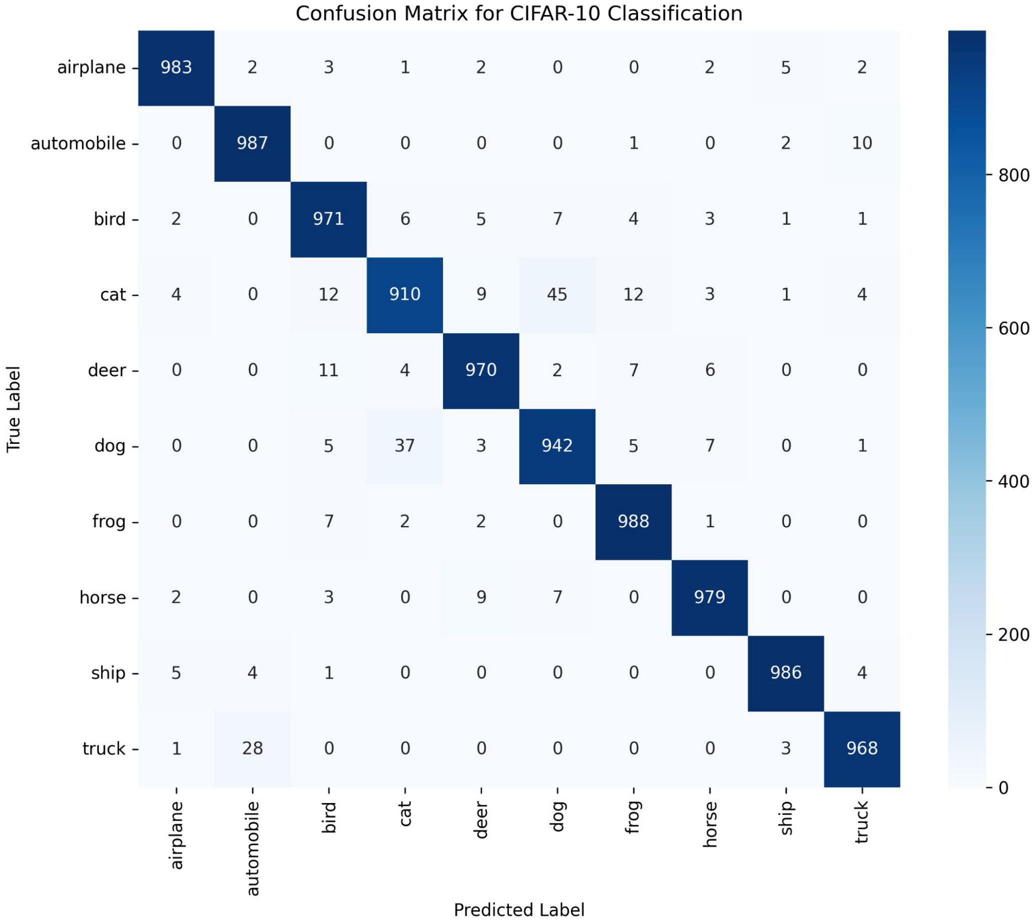
Training vs Validation Loss



- Sharp fall in loss during early epochs
- Validation loss consistently decreasing
- No oscillation → stable training
- Dropout + batch normalization preventing overfitting
-

Final Model Evaluation

The matrix shows **high precision and recall** across all classes, with only minor confusion occurring between visually similar objects (e.g., cat/dog, truck/automobile).



Model Comparison

Model	Train acc(%)	Valid acc(%)	Test acc(%)	Train Loss	Valid Loss
CNN	78.2%	68.6%	68.6%	0.62	0.997
CNN Aug.	71.0%	76.1%	76.8%	0.84	0.691
MobileNetV2	95.7%	91.2%	91.2%	0.13	0.267
EfficientNetB0	98.3%	96.7%	96.7%	0.04	0.1
DenseNet121	98%	94.2%	94%	0.06	0.021
EfficientNetv2B0	98%	97%	97%	0.057	0.096

Transfer Learning vs. Scratch

Feature	Transfer Learning (TL)	Building From Scratch
Data Requirement	Low (Small to medium dataset)	High (Requires massive dataset, often millions)
Training Time	**Very Fast (Only fine-tuning the last layers)**	**Very Slow (Training all layers from scratch)**
Computational Cost	Low (Primarily inference and fine-tuning)	**Very High (Requires extensive GPU/TPU time)**
Initial Performance	**High Baseline (Leverages pre-trained knowledge)**	Low (Starts randomly, depends entirely on your data)
Customization	Low (Constrained by the base architecture)	**High (Full control over every layer)**
Best For	Common tasks, small/medium data, quick deployment	Novel problems, highly unique domains, unlimited resources

Conclusion

The transfer learning models, particularly **EfficientNetv2B0** and **EfficientNetB0**, are the most powerful, high-accuracy, production-ready image classifiers, **significantly outperforming** the custom CNN and the MobileNetV2 model.

EfficientNetv2B0 achieved the highest test accuracy (97%), narrowly exceeding EfficientNetB0's 96.7% and DenseNet121's 94%.