Neural Network Models for Object Recognition

A Deep Learning Approach for CIFAR-10 Classification

Mariam Almarzooqi

The Challenge: Automated Object Recognition

Applications: Self-driving cars, security, medical imaging.

Dataset: CIFAR-10 (60,000 32x32 color images, 10 classes).

My Approach: Track 2 - Deep Learning

Focus: Convolutional Neural Networks (CNNs).
Strategy: Transfer Learning.
Why? State-of-the-art performance and efficiency.

Preparing the Data for Training

Total Data: 60,000 images.
Training Set: 45,000 images (for learning).
Validation Set: 5,000 images (for tuning).
Test Set: 10,000 images (for final evaluation).

Rationale: Why Validation is Crucial

Acts as a "practice exam" during training.
Provides unbiased performance feedback.
Key purpose: To detect and prevent overfitting.

Strategy: Transfer Learning with MobileNetV2

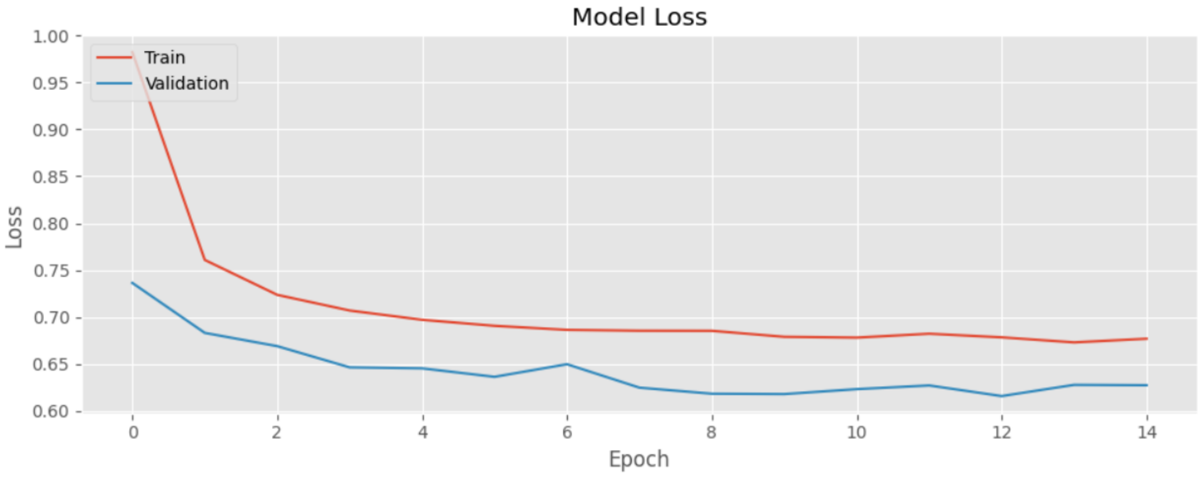
Leverages a powerful model pre-trained on ImageNet.
Freeze the pre-trained layers to keep their knowledge.
Train only a new, small classifier head.
Result: High accuracy with minimal training time.

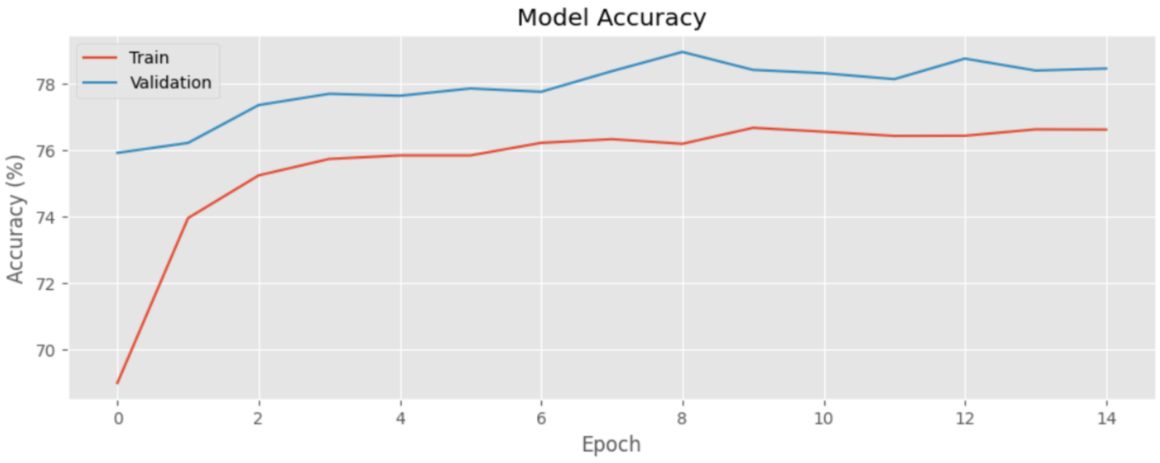
Model Design Decisions

Parameter	Value
Base Model	MobileNetV2
Optimizer	Adam
Loss Function	Cross-Entropy
Learning Rate	0.001
Batch Size	64
Epochs	15

Learning Curves: Accuracy & Loss

- Accuracy consistently increased, loss decreased.
- No signs of significant overfitting.
- Successful learning and generalization.





Final Evaluation: Test Accuracy of 78%

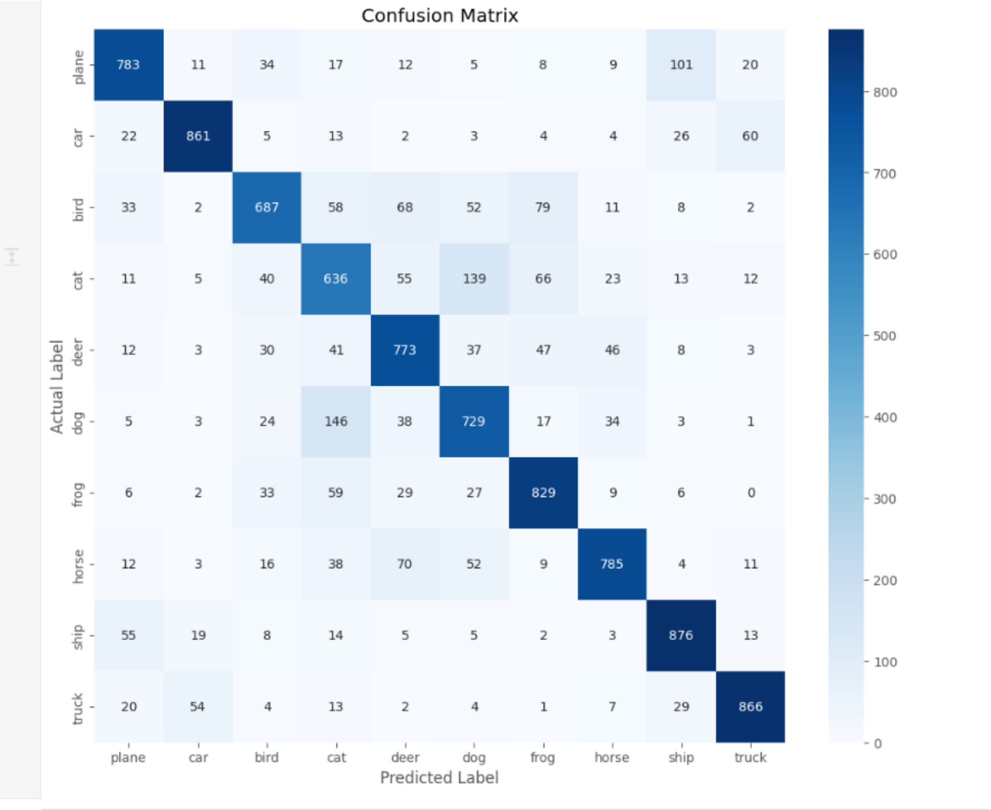
- **Strongest Classes:** Car, Truck, Ship (>84% F1-score).
- **Weakest Classes:** Cat (63% F1-score), Dog (71% F1-score).

Classification Report:

	precision	recall	f1-score	support
plane	0.82	0.78	0.80	1000
car	0.89	0.86	0.88	1000
bird	0.78	0.69	0.73	1000
cat	0.61	0.64	0.63	1000
deer	0.73	0.77	0.75	1000
dog	0.69	0.73	0.71	1000
frog	0.78	0.83	0.80	1000
horse	0.84	0.79	0.81	1000
ship	0.82	0.88	0.84	1000
truck	0.88	0.87	0.87	1000
accuracy			0.78	10000
macro avg	0.78	0.78	0.78	10000
weighted avg	0.78	0.78	0.78	10000

Analysis of Model Errors

- Strong diagonal confirms high overall accuracy.
- **Key Issue:** Significant confusion between 'cat' and 'dog'.
- Minor confusion between vehicle types ('car' vs 'truck').

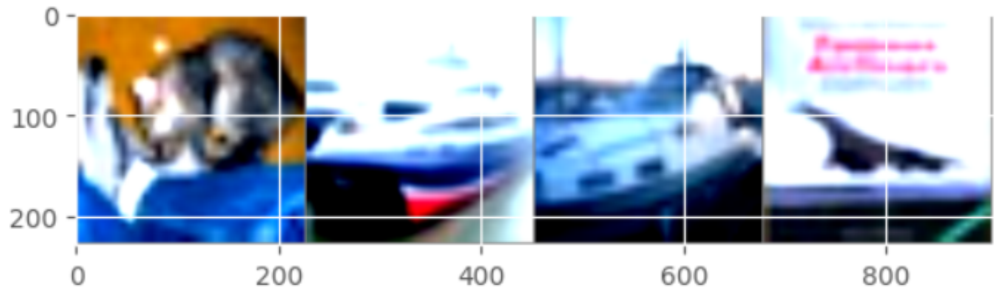


Model Demonstration in Action

- **Ground Truth:** cat, ship, ship, plane
- **Predicted:** dog, ship, ship, plane
- Perfectly illustrates the 'cat vs dog' challenge.

```
print('Predicted:   ', ' '.join(f'{CLASSES[predicted[j]]:5s}' for j in range(4)))
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for inte
```




```
GroundTruth:  cat   ship  ship  plane
Predicted:    dog   ship  plane plane
```

How Does This Approach Compare?

APPROACH	STRENGTHS	WEAKNESSES
Track 2 (Ours)	High accuracy, fast development time	"Black box" interpretability, computationally heavy to train
Track 1 (Classical)	More interpretable, less compute power needed	Lower accuracy, requires complex feature engineering
Track 3 (Advanced)	Potentially SOTA accuracy, can discover novel architectures	Very complex, extremely high computational cost and time

Conclusion

- **Successfully trained a model with 78% accuracy.**
- **Key Takeaway:** Transfer learning is a powerful and efficient strategy for computer vision.
- **Limitations:** Model struggles with low-resolution, visually similar classes.