AI-Powered Image Classification System

Milan Harishbhai Soni

Date:

1. Project Objective:

The goal of this project was to design, train, and evaluate a Convolutional Neural Network (CNN) to classify images from the CIFAR-10 dataset. This report details the data pipeline, the custom model architecture, and the final performance results.

2. Data Processing:

Dataset: I chose the CIFAR-10 dataset.

- **Splitting:** As needed, I combined the dataset and re-split it into a **70% / 15% / 15%** ratio:
  - **Training Set:** 42,000 images
  - **Validation Set:** 9,000 images
  - **Testing Set:** 9,000 images
- **Normalization:** All pixel values were scaled from 0-255 to 0.0-1.0.
- **Label Encoding:** Labels were one-hot encoded for the model.

3. Model Architecture:

I built a custom CNN from scratch based on the assignment's core tasks. The architecture includes three main convolutional blocks:

- **Input:** (32, 32, 3)
- **Block 1:** Conv2D (32 filters), BatchNormalization, MaxPooling
- **Block 2:** Conv2D (64 filters), BatchNormalization, MaxPooling
- **Block 3:** Conv2D (128 filters), BatchNormalization, MaxPooling
- **Classifier:** Flatten, Dense(128), Dropout(0.5), Dense(10, 'softmax')

This design includes all required components: three convolutional layers, BatchNormalization, and Dropout.

4. Training and Optimization:

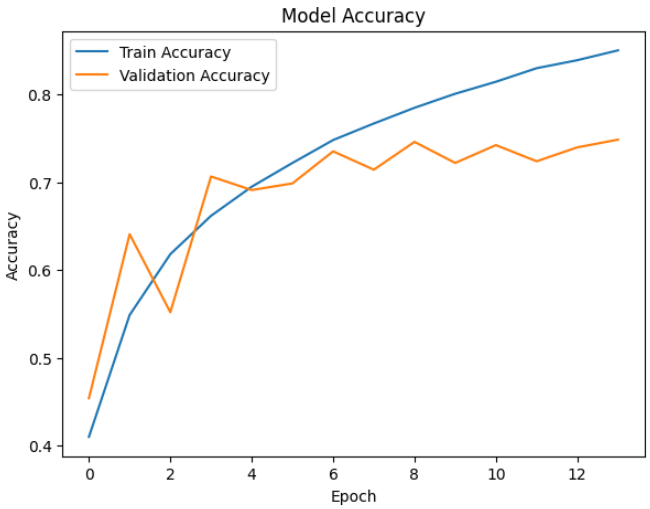
- **Optimizer:** Adam
- **Loss Function:** categorical_crossentropy
- **Optimization:** I used EarlyStopping to monitor validation loss and ModelCheckpoint to save the best-performing model. This prevents overfitting.

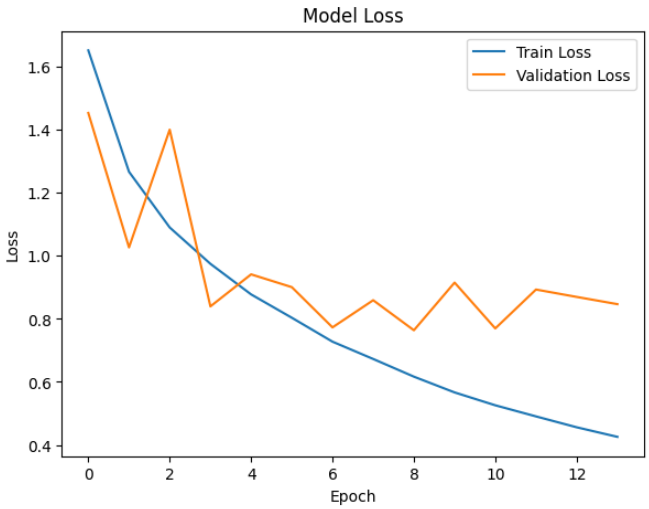
5. Final Test Results:

The best model was loaded and evaluated on the 9,000-image test set.

- **Final Test Accuracy: 75.02%**
- **Final Test Loss: 0.760**

Training and Validation Curves:

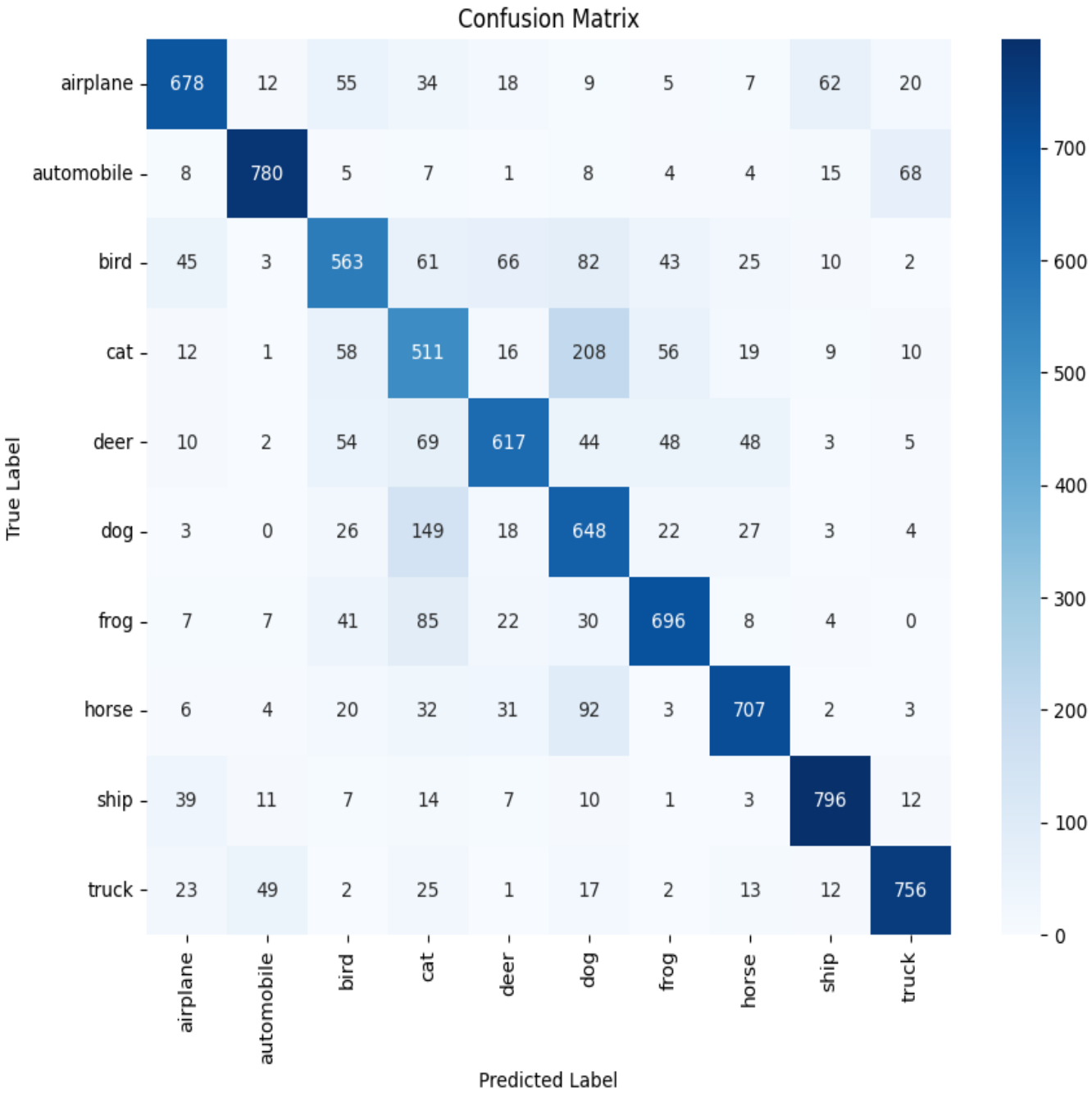




Classification Report:

	precision	recall	f1-score	support
airplane	0.82	0.75	0.78	900
automobile	0.90	0.87	0.88	900
bird	0.68	0.63	0.65	900
cat	0.52	0.57	0.54	900
deer	0.77	0.69	0.73	900
dog	0.56	0.72	0.63	900
frog	0.79	0.77	0.78	900
horse	0.82	0.79	0.80	900
ship	0.87	0.88	0.88	900
truck	0.86	0.84	0.85	900
accuracy			0.75	9000
macro avg	0.76	0.75	0.75	9000
weighted avg	0.76	0.75	0.75	9000

Confusion Matrix:



Sample Predictions:

Key Challenges and Improvements:

The main challenge was stopping the custom CNN from overfitting. It was needed to use BatchNormalization after each convolution and Dropout in the classifier. EarlyStopping also played a crucial role. It halted the training at the best epoch, which was around 14, before performance started to decline.