Assignment Two

Basira Daqiq | Assignment Github Repo

Task 1: Model Architecture

We trained a CNN to classify 10 object categories from the CIFAR-10 dataset. The dataset contains 60,000 images across the following classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck — with 6,000 images per class. The dataset was loaded using torchvision.datasets, which provides a pre-split version of 50,000 training images and 10,000 test images. The training set was further split using a random sampler, reserving 20% (10,000 images) for validation, and 40,000 images for training.

The following model architecture was designed to meet the project requirement of approximately 70% test accuracy. The model consists of three 2D convolutional layers, each followed by a max pooling layer. The first convolutional layer accepts a 3-channel (RGB) input and outputs 16 feature maps; the second takes 16 channels as input and outputs 32; and the third takes 32 channels and outputs 64. Each convolutional layer uses a 3×3 kernel with padding of 1 to preserve spatial dimensions, and each max pooling layer uses a 2×2 window, halving the spatial dimensions at each stage. Starting from a 32×32 input, the feature maps are reduced to 16×16, then 8×8, then 4×4 after the three pooling layers, yielding a tensor of shape 64×4×4.

The output of the final pooling layer is passed through a flatten operation, which reshapes the 64×4×4 tensor into a 1,024-dimensional vector. This is necessary because the subsequent fully connected (linear) layers expect one-dimensional input. The first linear layer maps the 1,024-dimensional input to 100 features, and the second linear layer maps those 100 features to 10 outputs — one for each target class. These 10 output values represent the model's raw scores (logits) for each class which is analogous to the probability of the sample being that class. The class with the highest score is selected as the final prediction. The Dropout with a probability of 0.25 is applied after the flatten step and after the first linear layer to reduce overfitting.

The model was trained for 30 epochs, but analysis of the training and validation loss curves revealed that overfitting began to occur at around 13–15 epochs as shown below:

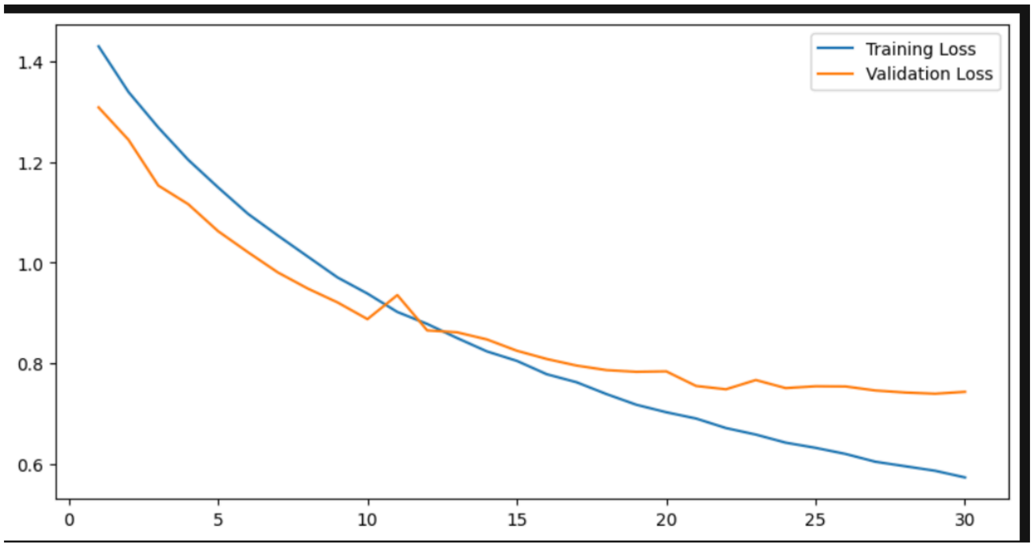


Figure 1. Training and validation accuracy vs number epochs for 30 epochs

This was evident from the training loss continuing to decrease while the validation loss began to rise, a classic sign of overfitting to the training data. To address this, the model was retrained for 13 epochs.

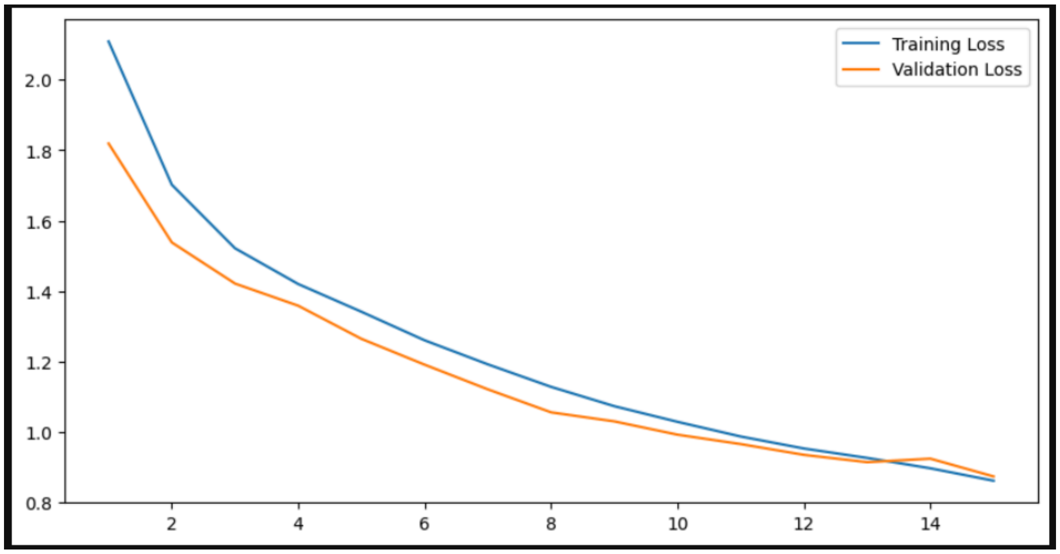
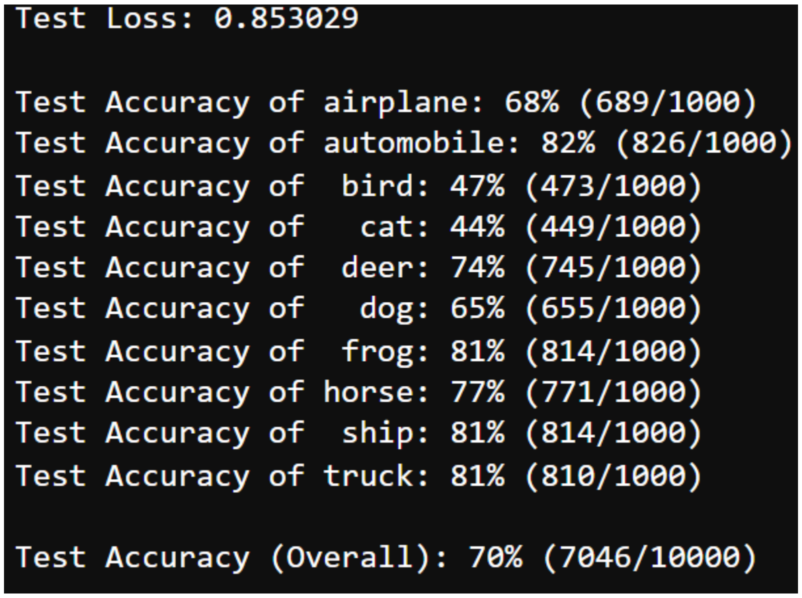


Figure 2. Training and validation accuracy for 15 epochs

The retrained model achieved an overall test accuracy of approximately 70%. Among the individual classes, ship and truck achieved the highest accuracy at around 81%, while bird and cat achieved the lowest at approximately 44% and 47%, respectively as shown in Table 1 below. This disparity is likely because ships and trucks have relatively simple, geometric shapes that are easier for the model to learn, whereas birds and cats exhibit more complex and variable silhouettes and textures that are harder to distinguish.

Table 1:



```
Test Loss: 0.853029

Test Accuracy of airplane: 68% (689/1000)
Test Accuracy of automobile: 82% (826/1000)
Test Accuracy of  bird: 47% (473/1000)
Test Accuracy of   cat: 44% (449/1000)
Test Accuracy of  deer: 74% (745/1000)
Test Accuracy of   dog: 65% (655/1000)
Test Accuracy of  frog: 81% (814/1000)
Test Accuracy of horse: 77% (771/1000)
Test Accuracy of  ship: 81% (814/1000)
Test Accuracy of truck: 81% (810/1000)

Test Accuracy (Overall): 70% (7046/10000)
```

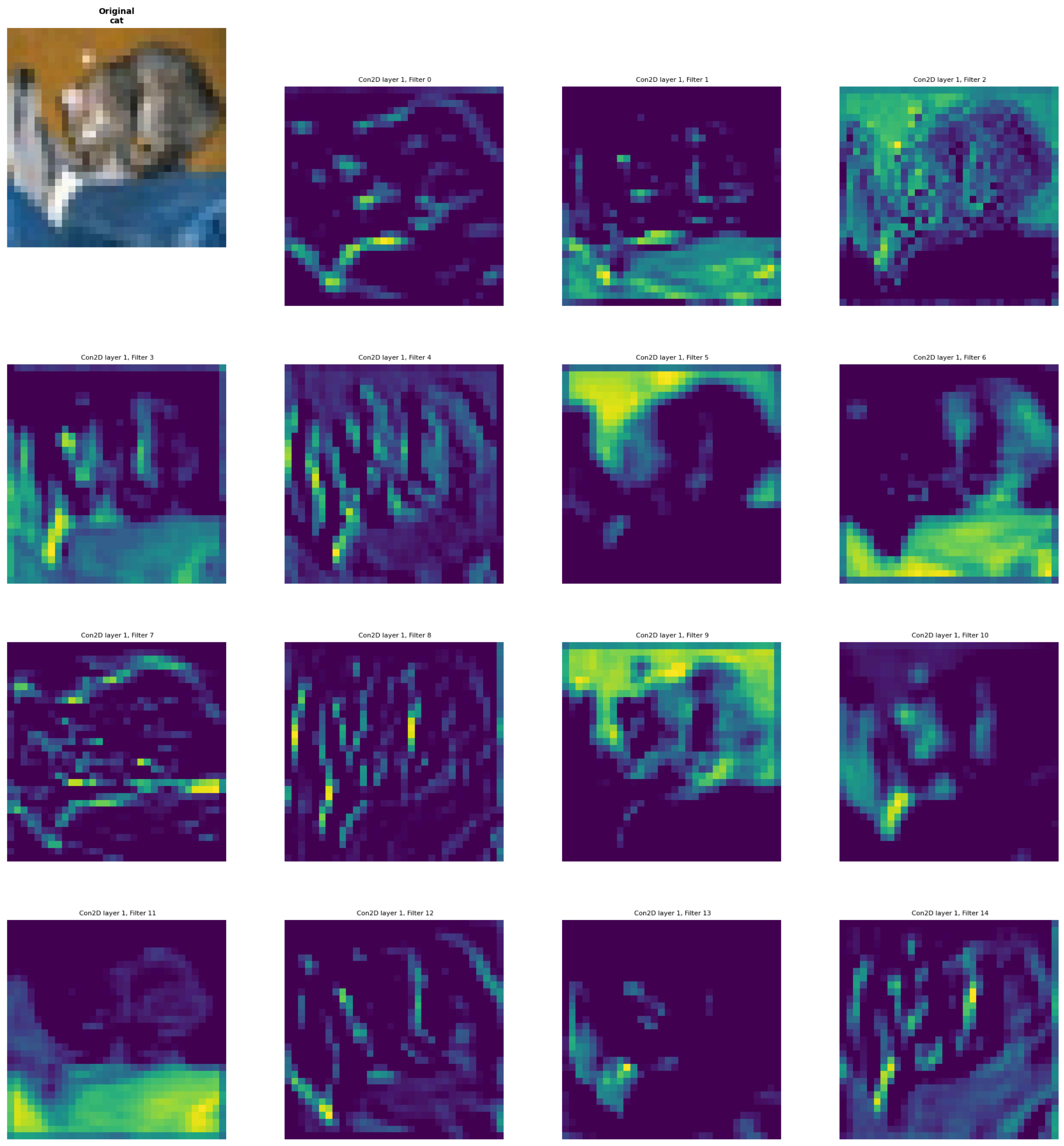
Task 2:

Part A:

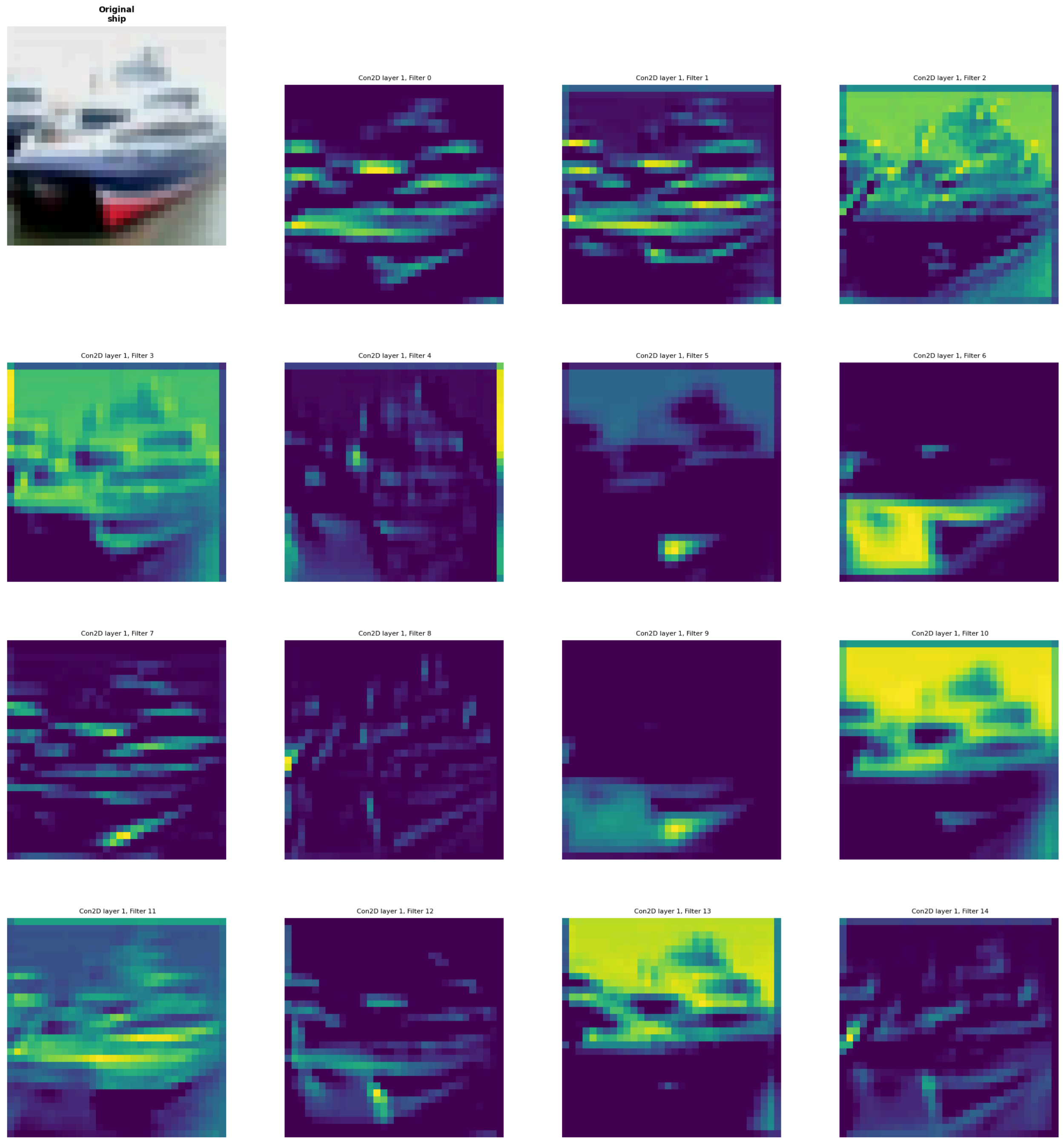
The figure shows feature maps for sample images of a cat, ship, and airplane. In each grid, the first image is the original input image, followed by 15 feature maps produced after a single convolutional layer and ReLU activation. Although we cannot determine the exact function of each filter with complete certainty, we can make reasonable interpretations based on the activation patterns.

Filter 0 appears to focus on edges, as indicated by the yellow regions representing higher activation values. Filter 2 appears to be activated primarily by brighter background regions, indicating sensitivity to large areas of high intensity or broad brightness gradients.

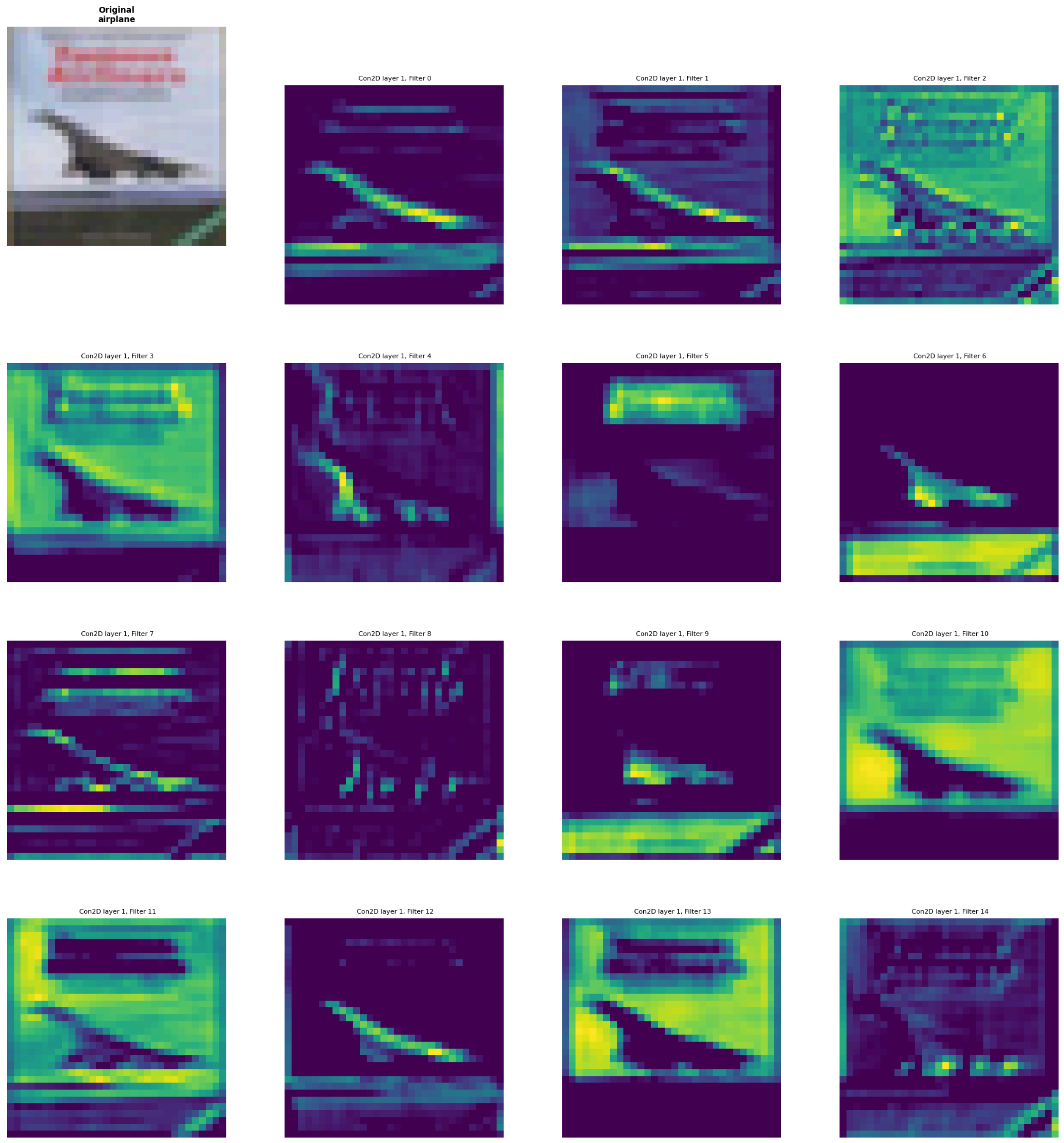
Filter 4 seems to respond strongly to textured areas within the image, suggesting it may be detecting fine patterns or surface details such as the body of the cat.



Feature Map 2 Ship



Feature Map 3 Airplane



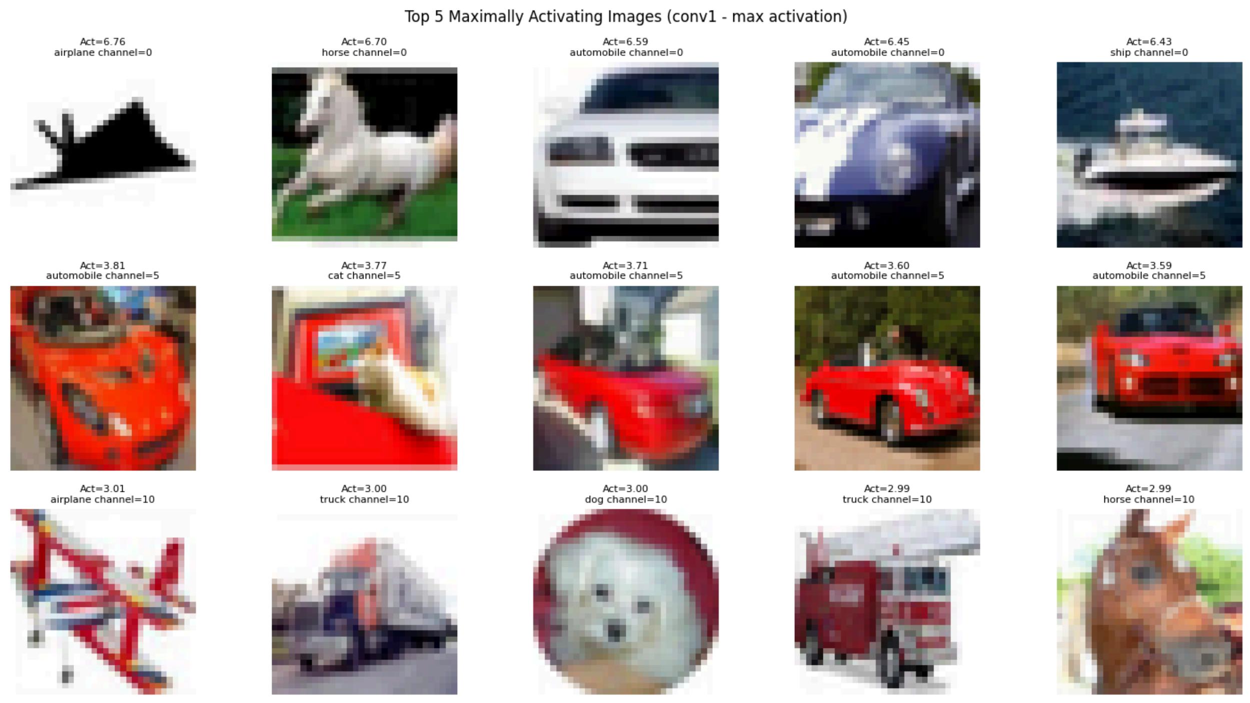
Part B

The figure shows the top five maximally activating images for Channels 0, 5, and 10 in the first convolutional layer.

Channel 0 appears to respond to strong edges and object boundaries, as many of its highest activations include airplanes, cars, and ships with clear contours and high contrast.

Channel 5 is primarily activated by red regions, especially red cars, indicating that it functions mainly as a red color detector.

Channel 10 seems to respond to warm red and brown tones along with structured object regions, as seen in images of airplanes, trucks, dogs, and horses with reddish or brown coloring.


Top 5 Maximally Activating Images (conv1 - max activation)
Act=6.76 airplane channel=0
Act=6.70 horse channel=0
Act=6.59 automobile channel=0
Act=6.45 automobile channel=0
Act=6.43 ship channel=0
Act=3.81 automobile channel=5
Act=3.77 cat channel=5
Act=3.71 automobile channel=5
Act=3.60 automobile channel=5
Act=3.59 automobile channel=5
Act=3.01 airplane channel=10
Act=3.00 truck channel=10
Act=3.00 dog channel=10
Act=2.99 truck channel=10
Act=2.99 horse channel=10