Alexander Baron

# CSCI-611 A2: CNN and Visualization

## Task 1: CNN Design and Training

This task involved designing and training a Convolutional Neural Network (CNN)  for the purpose of image classification. The network includes three convolutional layers with progressively increasing filter depths of 32,64, and 128 filters. Each convolutional layer employs a kernel size of 3x3 and a padding of 1, which helps preserve the spatial dimensions of the input feature maps across the convolutional layers.

Following every convolutional layer, Batch Normalization was applied with a number of features matching the output channels of the preceding layer ( 32, 64, and 128). Batch Normalisation stabilises the distribution of activations during training, accelerating convergence. A Rectified Linear Unit (ReLU) activation function is applied after each convolutional layer, introducing non-linearity and enabling the network to learn complex feature representations.

Spatial downsampling is achieved through a single Max Pooling layer using a 2x2 kernel with a stride of 2. This halves the height and width of the feature maps, reducing computational costs. Two dropout layers are incorporated to mitigate overfitting. Initially, both layers were assigned a dropout of 0.25, which would randomly deactivate 25% of neurons during each forward pass. The feature maps are passed to two fully connected layers that constitute the classification head. These layers transform the high-dimensional feature representations into class logits, which are then converted into class probabilities.

Cross Entropy Loss was selected as the loss function for this training, which is the standard choice for multi-class classification tasks. Cross-entropy measures the divergence between the predicted class probability distribution and the true label distribution. Training was initially performed using Stochastic Gradient Descent (SGD). SGD updates model parameters using the gradient computed from a single batch at each step. Following the SGD experiments the optimiser was switched to adam to assess wether its adaptive learning rate mechanism could yield further improvements.

Two learning rates were evaluated under the SGD optimiser. An initial learning rate of 0.01 was tested first; this configuration resulted in slow convergence followed by overfitting. Yielding a Training Loss of 0.250 and a Validation Loss of 0.678. The learning rate was increased to 0.1, which produced noticeable improvements in model accuracy. Although a higher learning rate can sometimes destabilise training, in this case, the larger update steps appear to help the model escape suboptimal regions of the loss landscape more effectively under SGD.

Training was initially conducted over 25 epochs. Once the ooverfitting issue observed in early experiments had been addressed through adjustments to the learning rate and regularaisation, the number of epochs was extended to 50 to allow the model sufficient time to converge more fully. A batch size of 20 was used throughout, meaning the model weights are updated after every 20 training samples.

The dataset used for this assignment was the CIFAR-10 dataset. The CIFAR-10 dataset consists of 60,000 color images of size 32x32 pixels across 10 mutually exclusive tasks (e.g., aeroplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck) with 6,000 images per class. The dataset is split between 50,000 training images and 10,000 test images.

To evaluate the impact of data augmentation on model generalisation, the model was trained and assessed both with and without augmentations applied. Augmentations were exclusively applied to the training set; the validation and test sets received no augmentation, ensuring that evaluation metrics reflect performance on unmodified data.

Three augmentation techniques were employed. First, RandomHorizontalFlip(), randomly mirrors each training image horizontally with a default probability of 0.5. Second, RandomCrop(32, padding=4) pads each image by 4 pixels on each side before randomly cropping back to the original 32x32 resolution. Third, ColorJitter(brightness=0.2, contrast=0.2, saturation=0.2) randomly perturbs the brightness, contrast, and saturation of each image within a range of ±0.2.

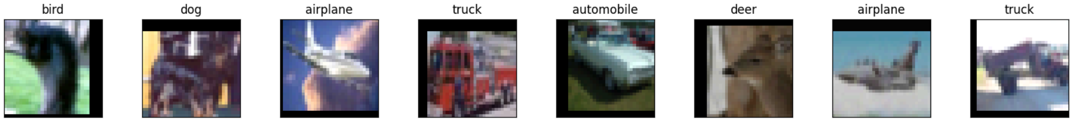

Figure 1.1: Augmentations of CIFAR-10

## Before Augmentations

The first training run was conducted using SGD with a learning rate of 0.02, a dropout rate of 0.25, 25 epochs, and no data augmentations. During the initial epochs, the model demonstrated consistent improvement in both training and validation loss, with no early signs of overfitting.

After reaching its peak at epoch 6. However, the validation loss began to increase while the training loss continued to decrease, a classic indicator of overfitting. The model was memorising the training data rather than learning generalisable features. The best validation loss achieved in this run was 0.678 at epoch 6/25, with a training loss of 0.250. The significant gap between the two values further confirms the degree of overfitting present.

Despite the overfitting observed during training, the model achieved a respectable overall accuracy of 76% on the test set. Performance varied considerably across the ten CIFAR-10 classes. The highest performing classes were Ship (90%), Truck (86%), and Frog (83%), which likely benefit from their visually distinctive features and relatively consistent appearances across samples. The model struggles the most with Bird (58%), Dog (61%), and Cat (68%). These lower performing classes share common characteristics; they are all organic, deformable subjects with high inter-class visual variability. Cat and Dog particularly exhibit significant inter-class similarity, making them more inherently challenging to discriminate.

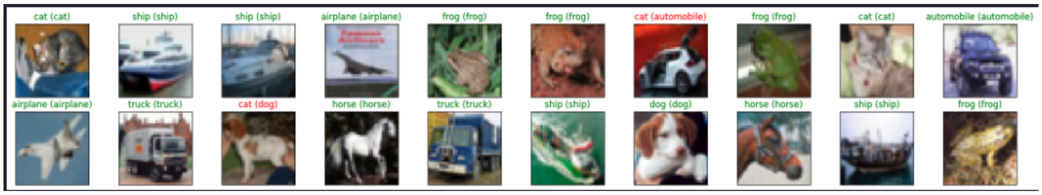

Figure 1.2: Example classification produced by the model

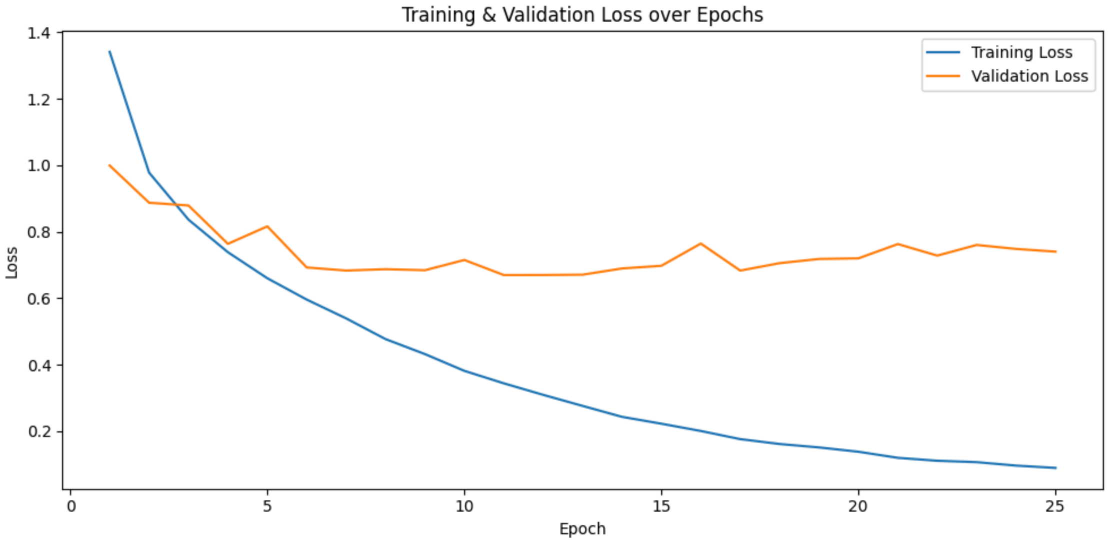

Figure 1.3 : Training and Validation Loss Curves

## After Augmentations

To evaluate the effect of data augmentation, the model was retrained under identical conditions to the baseline run, SGD Optimizer, learning rate of 0.01, and 25 epochs, with the sole difference being the application of RandomHorizontalFlip(), RandomCrop(32, padding=4), and ColorJitter(brightness=0.2, contrast=0.2, saturation=0.2) to the training data. The validation and test sets remained unaugmented.

The augmented model achieved a best validation loss of 0.598 and a training loss of 0.599 at the final epoch (25/25). The near-identical training and validation loss values indicate the augmentation was highly effective at suppressing overfitting. Overall accuracy improved to 80%, a 4% point gain over the non-augmented baseline.

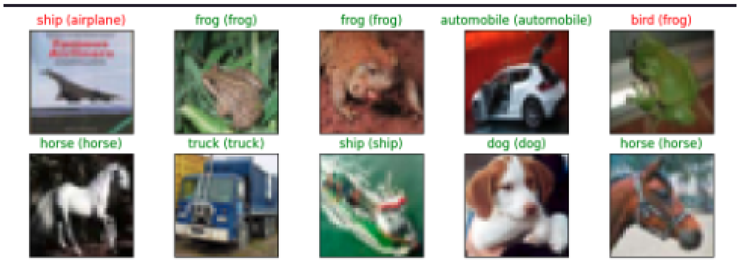

Figure 1.4: Example Classifications produced by the Augmented Model

Building on the observation that the augmented 25-epoch model had not yet fully converged, three adjustments were made simultaneously. The number of training epochs was double to 50, the learning rate was increased from 0.01 to 0.1to encourage the model to reach better loss values more consistently, and the dropout probability was raised from 0.25 to 0.35 to provide stronger regularization against the risk of overfitting the longer training run.

The model surpassed its previous best validation loss at epoch 28, demonstrating that the additional epoch was necessary to unlock further improvement. Training continued to progress beyond this point, ultimately achieving a best validation loss of 0.521 and a training loss of 0.552. The model achieved an overall accuracy of 83%, representing a 3% point improvement over the augmented 25 mobel and a cumulative 7% point gain over the original non-augmented baseline.

The strongest classes were Frog (93%), Truck (92%), and Automobile (91%), with Frog now surpassing vehicle classes to take the top spot, likely benefiting from its distinctive color, and texture features. The weakest class remained Cat (64%), Dog (73%), and Deer (75%), consistent with previous runs. Notably Cat saw a meaningful improvement from 52% in the previous run to 64% here, suggesting the stronger regularization and additional epochs helped the model better distinguish it from visually similar classes such as Dog.

Figure 1.5: Example Classifications

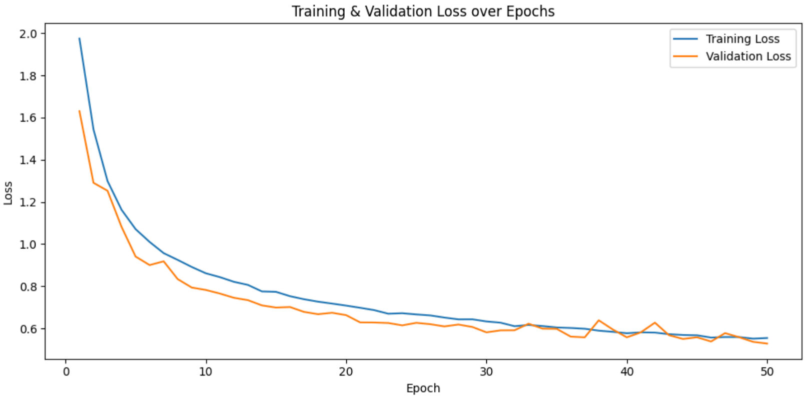

Figure 1.6: Training and Validation loss curves for the 50 epoch run

## Adding Adam Optimizer

Having established a strong baseline with SGD across three successive runs, the optimizer was switched to Adam to investigate whether its adaptive learning mechanism could push model performance further. While SGD applies a uniform learning rate to all parameters, Adam maintains individualized, adaptive learning rates for each parameter based on estimates of first and second order gradients, allowing it to converge faster and navigate loss landscapes that SGD can struggle with. All other hyperparameters were held constant to isolate the effect of the optimizer change.

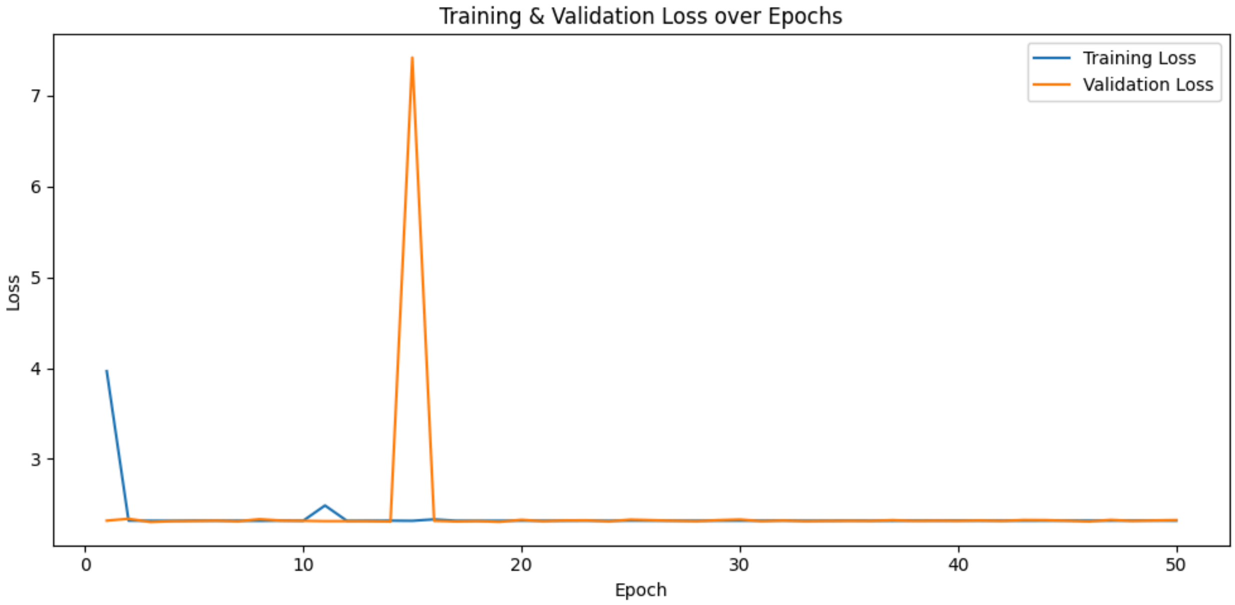

FIGURE 1.7: Training and Validation Loss Curves for Adam at lr = 0.1

As shown in Figure 1.7, the model neither overfitted nor improved meaningfully across all 50 epochs, instead becoming trapped in a plateau throughout training. This behavior is consistent with a learning rate that is too large for an optimizer that relies on momentum. A learning rate of 0.1 is considerably higher than Adam's conventional default of 0.001; this appears to have prevented any meaningful learning from taking place.

As a result, the model achieved a test accuracy of just 10%, equivalent to random chance across 10 classes, and was unable to correctly classify any images beyond a single class (Airplane). This demonstrates that Adam is highly sensitive to the choice of learning rate, and that a value appropriate for SGD can be entirely unsuitable for Adam, motivating a substantial reduction in learning rate.

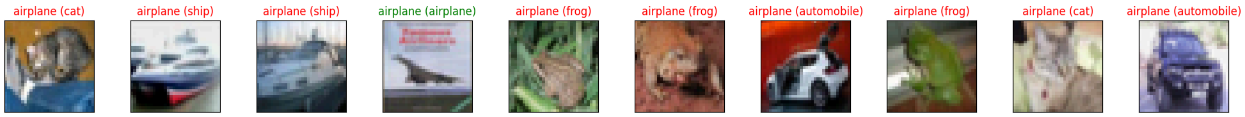

Figure 1.8 : Example classifications from Adam lr = 0.1 model

Figure 1.8 further illustrates this failure mode. Rather than distributing predictions across multiple classes, the model assigned every input image to the Airplane class, regardless of its true label. This is an example of a degenerate classifier, because Airplane makes up 10% of the CIFAR-10 test set, predicting it for every sample yields exactly 10% accuracy. The model did not learn any discriminative features; it simply collapsed to a single class output, which happened to be correct only for Airplane.

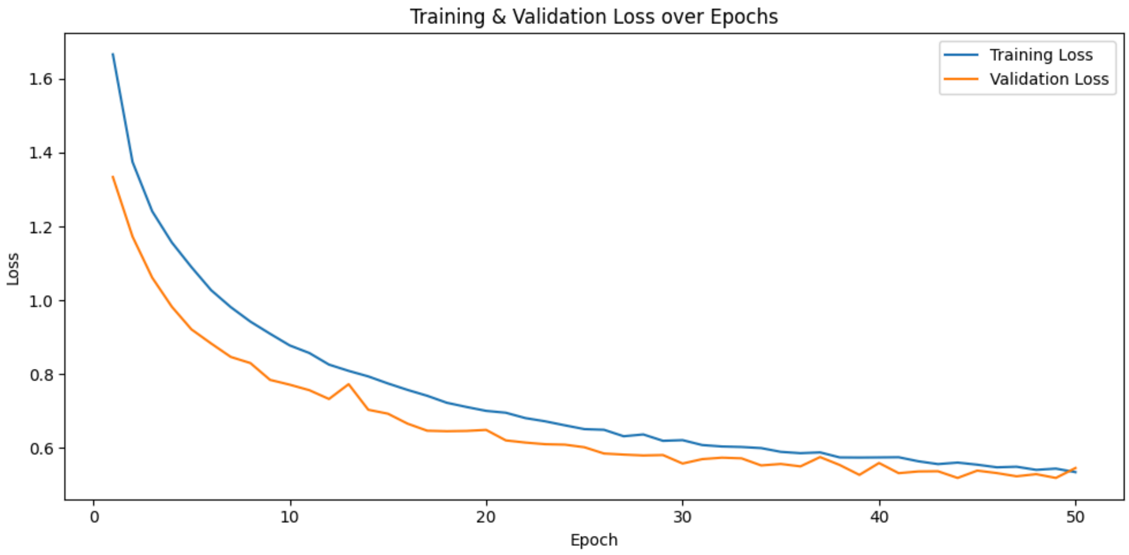

Figure 1.9: Training and Validation loss curves for the Adam lr = 0.001 model

Following the degenerate performance observed at lr = 0.1, the learning rate was reduced to 0.001, Adams recommended default. All other settings were kept constant. The reduction is learning rate proved highly effective. The model was able to navigate the loss landscape steadily without collapsing into a degenerate state, achieving a best validation loss of 0.5180 and a training loss of 0.5337. Overall accuracy reached 83%, matching the best SGD result from previous runs, and demonstrating that Adam, when configured with an appropriate learning rate, is capable of achieving comparable performance.

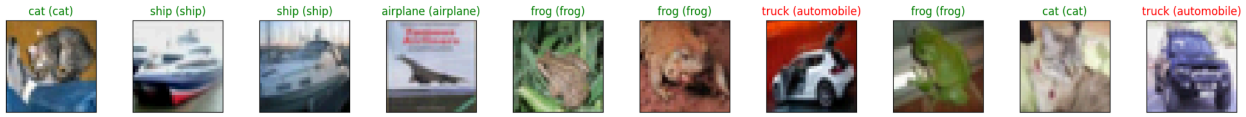

Figure 1.10: Example classifications produced by the Adam lr=0.001 model

# Task 2: Feature Map Visualization

## Feature Maps

This task focused on understanding what the CNN learns internally by visualizing the feature maps produced by the first convolutional layer. Three test images were selected from different classes { Automobile, Frog, and Ship}, and passed through the trained CNN from task 1. At least 8 feature maps per image were extracted from the first convolutional layer, each corresponding to the activation of a different learned filter.

Figure 2.1 shows the eight feature maps produced by the first convolutional layer for a test image of an Automobile, which the model correctly classified. Examining the activated regions across filters reveals that different filters have specialized to detect different structural aspects of the image.

Several filters appear to respond strongly to the curved contours of the car body, highlighting the vehicle's outline and bodywork. Other filters show concentrated activation in regions consistent with the wheels, suggesting the model has learned to detect circular, low-positioned structures as a distinguishing feature of the Automobile class.

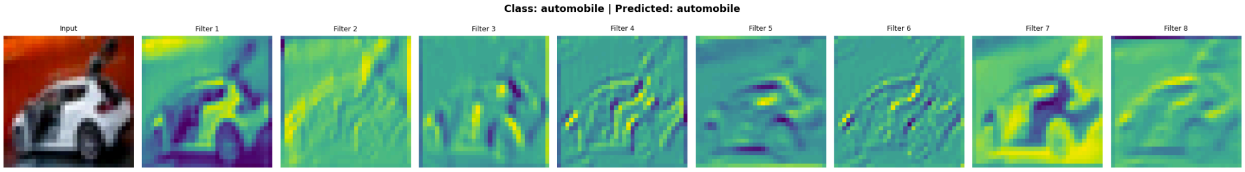

Figure 2.1: First convolutional layer feature maps for an automobile

The frog Frog class provides an interesting contrast to the Automobile. As shown in Figure 2.2, the majority of filters in the first convolutional layer appear oriented toward detecting textual patterns and organic shape characteristics of a frog's appearance. Filter 1 responds strongly to the body of the frog producing high activations across the regions where the frog's torso is located. Filter 8, the final filter appears to focus on detecting the elongated, angles structures consistent with the frog's legs.

The remaining filters respond to various surface patterns and directional edges, likely capturing the mottled skin texture and the surrounding foliage, which the model learns to separate from the subject itself. The filters suggest the model builds it Frog representation from a combination of body shape, limb structure, and surface texture cues.

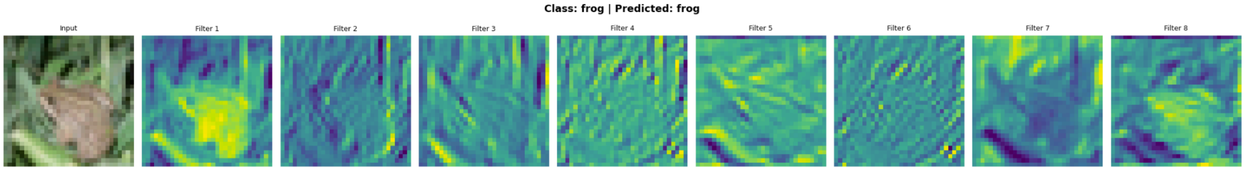

Figure 2.2: First convolutional layer feature maps for a Frog

The ship class reveals another distinct pattern of filter specialization as shown in Figure 2.3. The model correctly classified this image, and the feature maps offer clear insight into which visual cues the model relied on. Filter 2 shows strong activation along the lower hull region of the ship, suggesting it has learned to detect the distinctive painted pattern or color boundary typically appearing on the underside of a vessel. Filter 3 responds sharply to the corner of the board, capturing the angular edge where the hull meets the waterline or the vessel's bow.

The remaining filters capture broader directional edges and structural lines across the hull, collectively encoding the elongated, horizontally dominant shape that characterizes ships in the CIFAR-10 dataset. The contrast between the Ship's largely geometric, edge based activations and the Frog's texture driven responses illustrate how the same convolutional layer adapts its filters to respond to specific visual properties of each class.

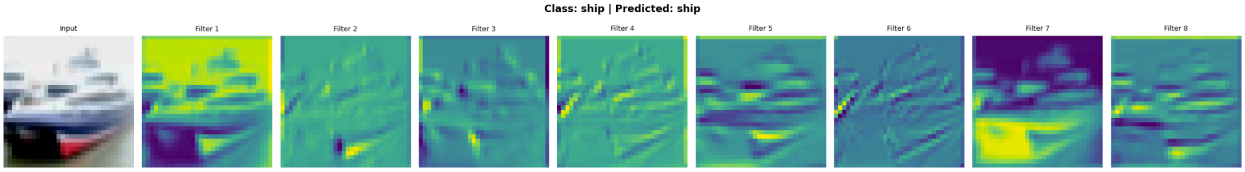

Figure 2.3: First convolutional layer feature maps for a Ship

## Maximally Activating Images

To further understand what individual filters in the first convolutional layer have learned to detect, the top 5 test imagesproducing the highest mean ReLU activation were identified for three selected filters: Filter 1, Filter 8, and Filter 16. Filter 1 is most strongly activate by images containing Airplanes and Birds with activation values ranging from 1.936 to 2.204

The common thread between these classes is apparent in Figure 2.5; both Airplanes and Birds are small, dark objects set against a bright, plain background. The filter appears to have specialised in detecting elongated shapes with a high contrast ratio against a light background.

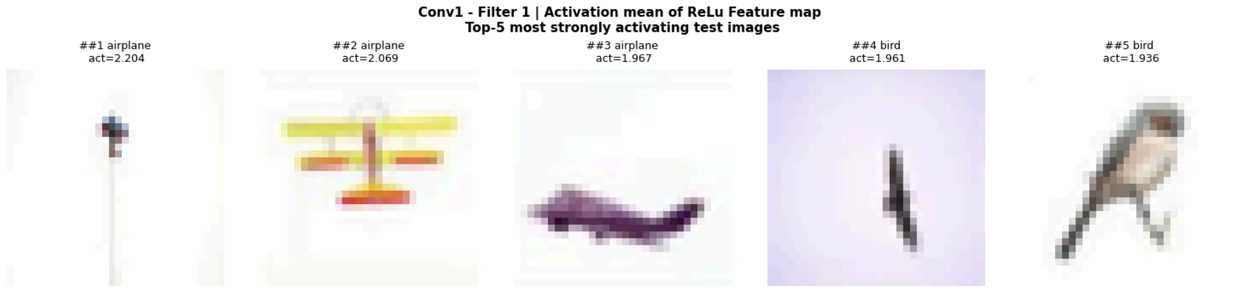

Figure 2.4: Top 5 test images producing the highest mean ReLU activation in Conv1 Filter 1

Filter 8 shows a strong preference for images of Airplanes against blue sky backgrounds with four of the top five activating images belonging to the Airplane class and one Automobile. Activation values are notably higher than Filter 1, ranging from 1.881 to 2.870. The shared visual property here is clearly the blue background evident in Figure 2.5, suggesting Filter 8 has learned to respond to the combination of a blue, high-saturation background, with a dark, structured foreground object. The inclusion of the Automobile is interesting, likely being activated due to the blue-toned background in the image rather than the vehicle itself. Confirming filter 8 is responding to the background color rather than the object class.

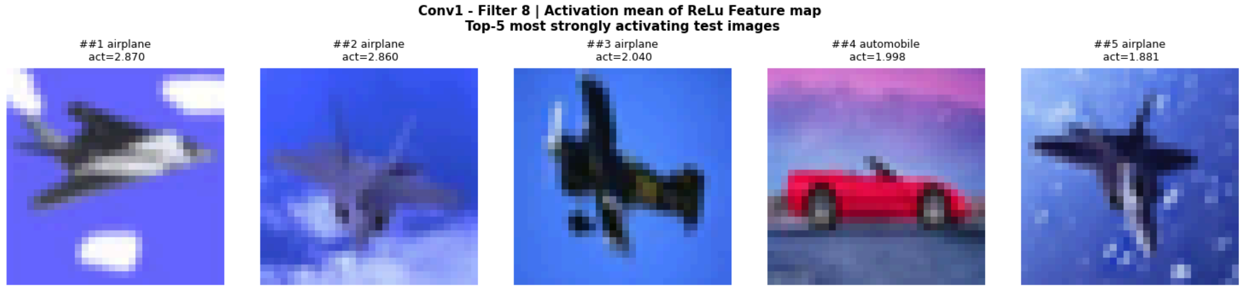

Figure 2.5 : Top 5 test images producing the highest mean ReLU activation in Conv1 Filter 8

Filter 16 displays the most dramatic activation of the three filters, with the top image reaching an activation of 10.905, significantly higher than the other filters. The images are once again dominated by the airplane class, with one Bird. All five images in Figure 2.6 share a vivid cyan or light blue background, and critically, all subjects are dark, wing-shaped objects set against a bright background. Tbjs suggest Filter 16 has become highly specialized; it fires strongly when it detects the specific combination laid out previously. The bird at rank 2 reinforces this interpretation; its dark triangular wing shape against a teal background closely mirrors the visual pattern of the Airplane images. This could be evidence of why the model conflates Bird and Airplane at the feature level.

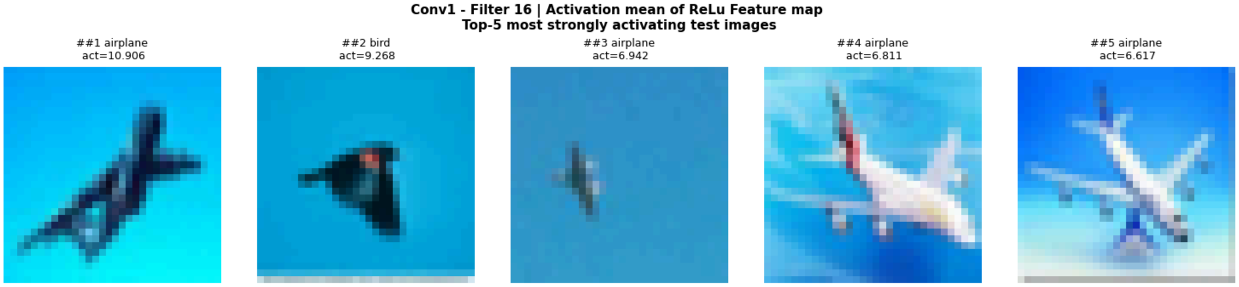

Figure 2.6: Top 5 test images producing the highest mean ReLU activation in Conv1 Filter 16