Applied Deep Learning - Assignment 4

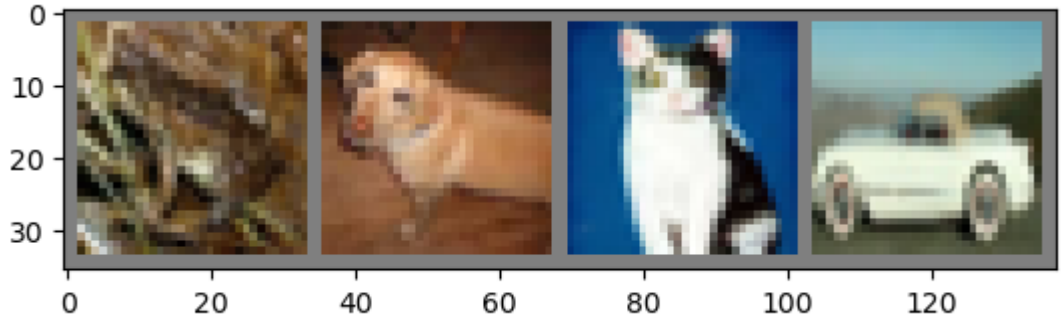
Task 1:

We followed the instructions from here, all of the code for this assignment is attached.

First we downloaded the CIFAR10 dataset.

The output of torchvision datasets are PILImage images of range [0, 1] so in order to normalize the data into range [-1,1]. we used transforms.normalize.

Here are a few training images:

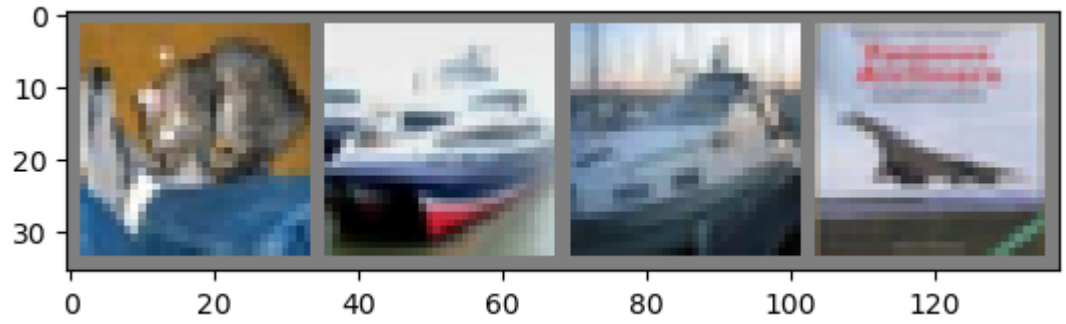


Next we defined a convolutional Neural Network with 2 convolutional layers and 3 linear fully connected layers for which the loss function is cross entropy and the optimizer uses SGD.

After that, we trained the network using 2 epochs counting the loss for each batch.

```
[1,  2000] loss: 2.246
[1,  4000] loss: 1.935
[1,  6000] loss: 1.711
[1,  8000] loss: 1.607
[1, 10000] loss: 1.536
[1, 12000] loss: 1.490
[2,  2000] loss: 1.423
[2,  4000] loss: 1.415
[2,  6000] loss: 1.376
[2,  8000] loss: 1.366
[2, 10000] loss: 1.325
[2, 12000] loss: 1.292
Finished Training
```

For demonstration we presented 4 images from the test data and their labels, also their prediction by the model:



The labels are :

[cat, ship, ship, plane]

The model's predictions:

[frog, plane, ship, plane]

Later, in order to see how the network performs on the whole dataset. We calculated the accuracy of our model using the predictions of the model for dataset for which we got:

Accuracy of the network on the 10000 test images: 54 %

Which is far better than chance which is 10%,

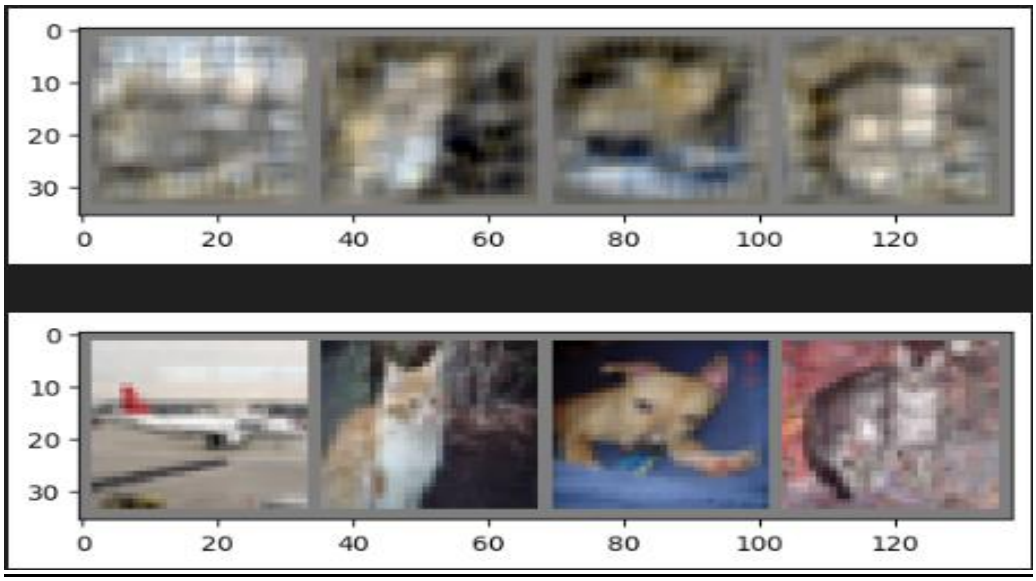
Meaning our model indeed learned something. Then we estimated what are the classes that performed well:

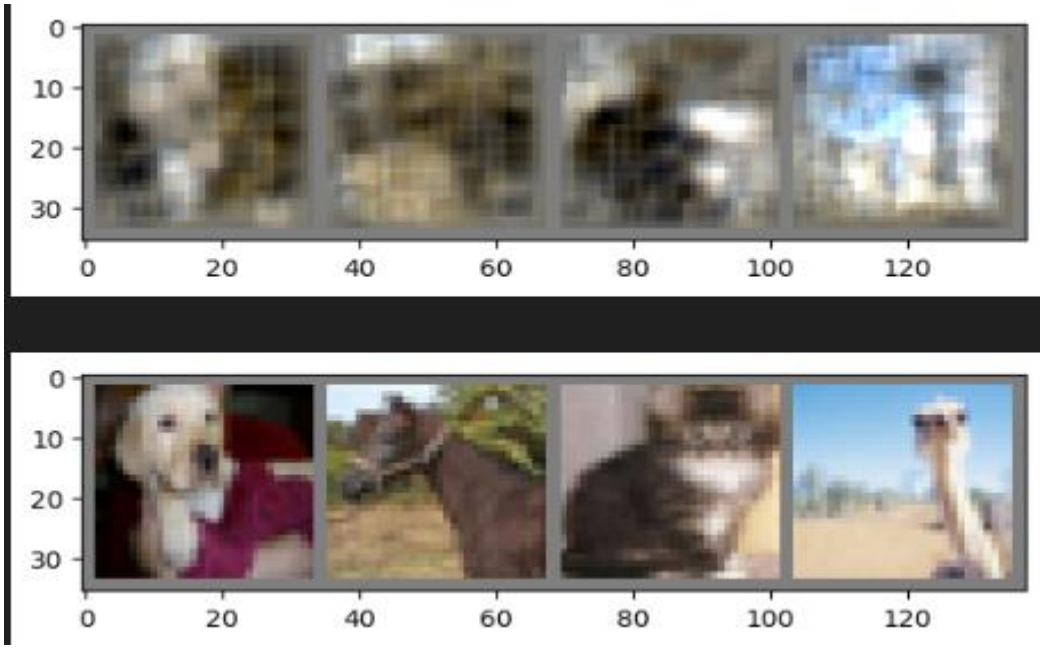
```
Accuracy for class: plane is 66.7 %
Accuracy for class: car   is 66.2 %
Accuracy for class: bird  is 25.0 %
Accuracy for class: cat   is 33.3 %
Accuracy for class: deer  is 37.9 %
Accuracy for class: dog   is 42.7 %
Accuracy for class: frog  is 75.8 %
Accuracy for class: horse is 57.4 %
Accuracy for class: ship  is 64.6 %
Accuracy for class: truck is 71.7 %
```

Task 2:

We modified our network from the previous task to also include two deconvolutional layers, which are able to reconstruct images after the convolutional process. As we can see in Figure 1 in the assignment guidelines, now the output of the network is $y, \hat{x}$, where $\hat{x}$ is the reconstructed image, made from the input image.

Some images and their reconstruction:



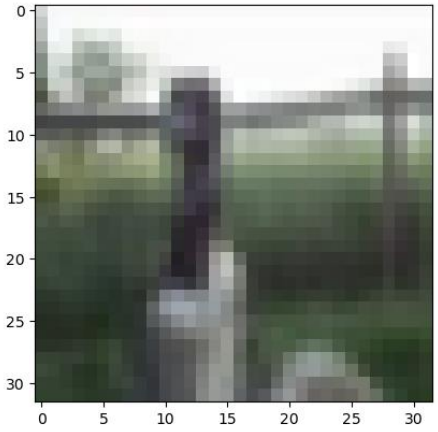


Accuracy of the network on the 10000 test images: 56.15%

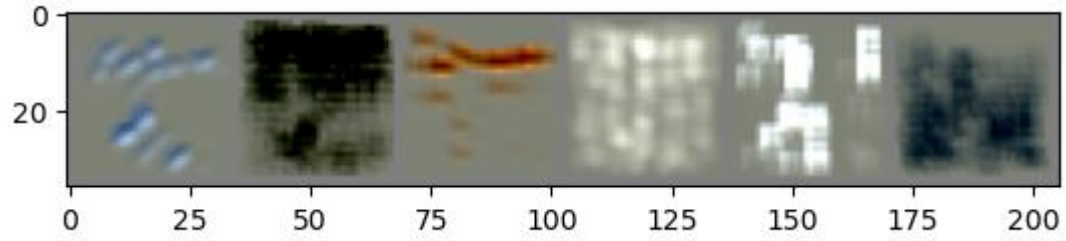
Task 3:

To do this task, we used two images: one from the train set, and another from the test set, and used our trained network from task2 to reconstruct each image, each time focusing on a different feature, while zeroing the other ones.

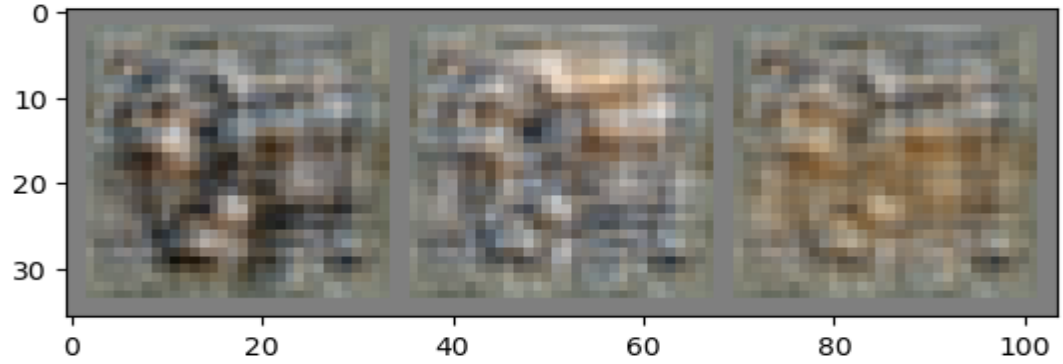
The training image we reconstructed from:



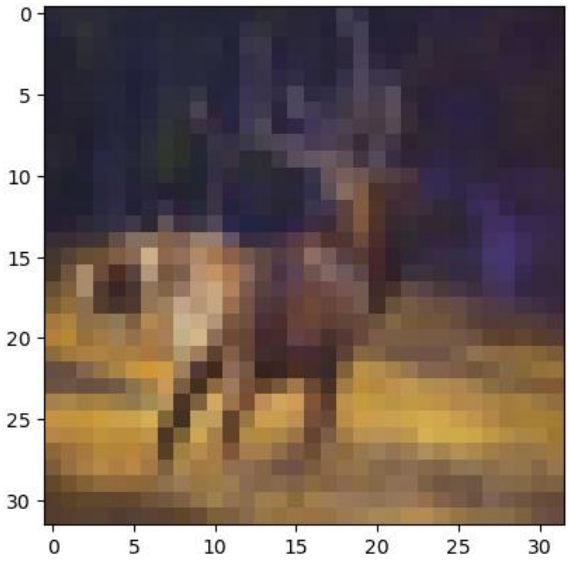
We took this training image, and after singling out each channel from z(1) which means zeroing out the other 5, we got:



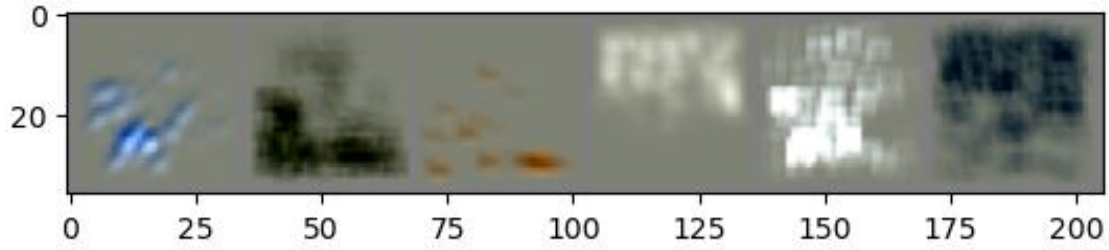
We took this training image, and after singling out 3 channels from z(2) which means zeroing out the other 15, we got:



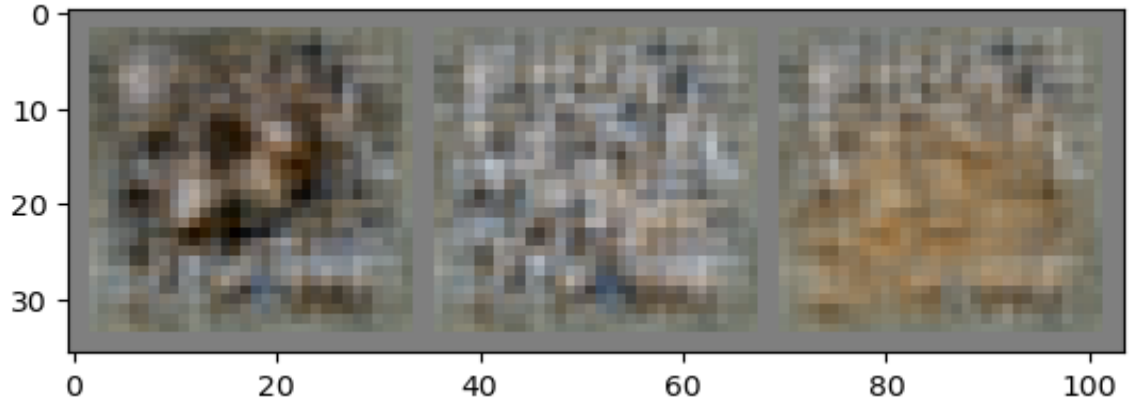
The testing image we reconstructed from:



We took this training image, and after singling out each channel from z(1) which means zeroing out the other 5, we got:



We took this test image, and after singling out 3 channels from z(2) which means zeroing out the other 15, we got:



As we can see, in the z(1) reconstruction, there are a few characteristics of the original image that each reconstructed image, that represents a single channel, is able to “capture”. while it is not definite and easy to identify, each channel focuses on different aspects of the original image, contributing to the multi-layered representation.
Some images, as we can see quite clearly in the first channel (the most left one), appear to focus on the edges and boundaries of the objects in the original image.
Other images were able to capture the shapes of the objects themselves or even, like channel 2 (the second from left) captured the color variations, providing information about the distribution of colors and transitions within the image.

Furthermore, it appears that the reconstructions of the features from z(2) are considerably more difficult to interpret, as there are no discernible patterns it is and a more condensed representation compared to the 6 channels. Due to the fact that this layer consists of 16 channels made up of z(1) rather than the original image, they are complex and difficult to perceive by the human eye. However, it still manages to capture some key characteristics of the original images, as we can see.

A CNN's kernels, which determine its features, are learned through iterative training. By optimizing the kernels using Stochastic Gradient Descent (SGD), meaningful features can be captured from the data. This, however, does not guarantee the creation of an image containing patterns that can be seen by humans based on the reconstruction of these features.