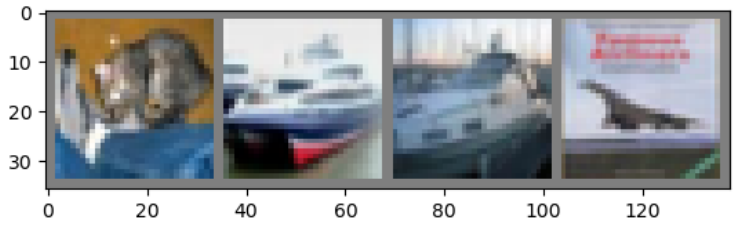
ADL – HW4

Task 1:



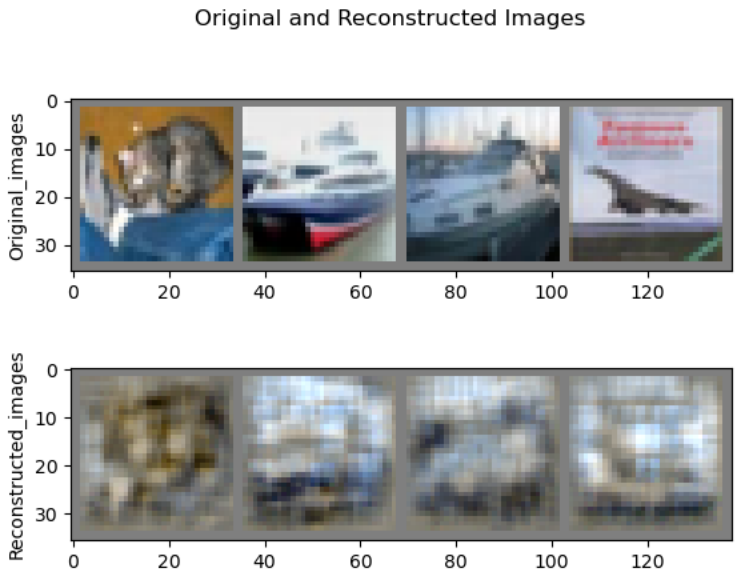
Ground Truth	Cat	Ship	Ship	Plane
Predicted	Cat	Ship	Ship	Ship

The model's accuracy of the network on the 10000 test images: 54 %

Class	**Accuracy**
plane	53.4 %
car	66.9 %
bird	45.8 %
cat	19.3 %
deer	39.7 %
dog	26.0 %
frog	80.0 %
horse	62.3 %
ship	81.3 %
truck	70.1 %

First, we defined a CNN as we saw in class, with two convolutional layers and three fully connected layers, with max pooling after each convolution and RELU as an activation function. Then we defined cross-entropy as a loss function and SGD as an optimizer function. We used a common pytorch loop to train the model for 2 epochs, and then we tested the model on the test data. we calculated the model's accuracy over the test data, and its accuracy rate for each independent class.

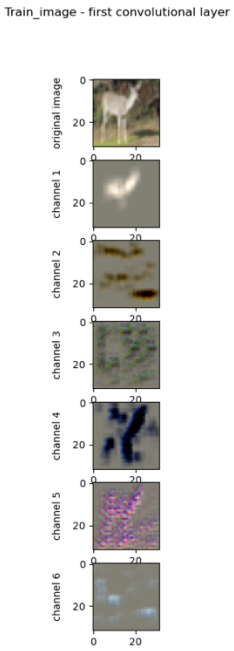
Task 2:

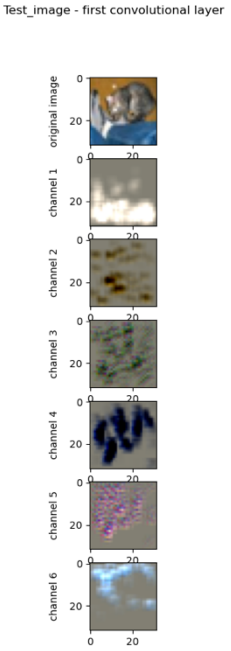


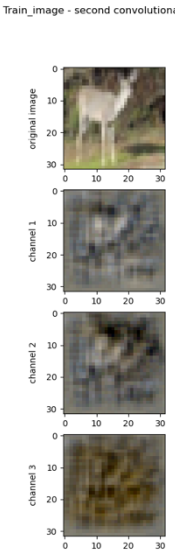
Accuracy of the network on the 10000 test images: 55 %

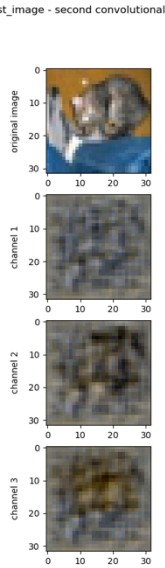
We used deconvolution and unpool layers to reconstruct the image. Although the reconstructed images are quite blurry, we can see the original images objects in them. The reason for the pictures blurriness might be that while the model learns the best way to reconstruct the images (depends on the lambda, for this example we used lambda = 1), the loss function also learns how to classify the images correctly. So, if we would increase the lambda in the loss function, we would expect better looking images, but a lower accuracy rate.

Task 3:


Train_image - first convolutional layer
original image
channel 1
channel 2
channel 3
channel 4
channel 5
channel 6


Test_image - first convolutional layer
original image
channel 1
channel 2
channel 3
channel 4
channel 5
channel 6


Train_image - second convolutional layer
original image
channel 1
channel 2
channel 3


Test_image - second convolutional layer
original image
channel 1
channel 2
channel 3

In Task 3 we took one image from the train data and one image from the test data and generated their reconstructed image for each channel after the first convolutional layer, and for each of the first three channels after the second convolutional layer. We can see that after the first layer, each channel represents deferent color or shade of the image (channel 1 white, channel 5 pink, channel 5 brown etc.), and for each channel you can spot outlines of different features of the original image. For example, in channel 1 of the train image, we can clearly spot the outlines of the deer. For the test image in channel 6, we can spot the outlines around the cat.
After the second convolutional layer, we could expect that each channel would represent more complex and abstract features of the image, but since the image's resolution is low to begin with and the second layer has 16 channels, it's hard to notice significant patterns for each channel. We can however notice vaguely different features of the original image on the different channels, for example in channel 3 of the test image we can notice the blue blanket at the bottom and the orange background, and for channel 1 of the train image we could spot the body of the deer in the middle of the picture. Perhaps if we took more channels in consideration for the reconstructed images, we could see more complex features of the images more clearly…