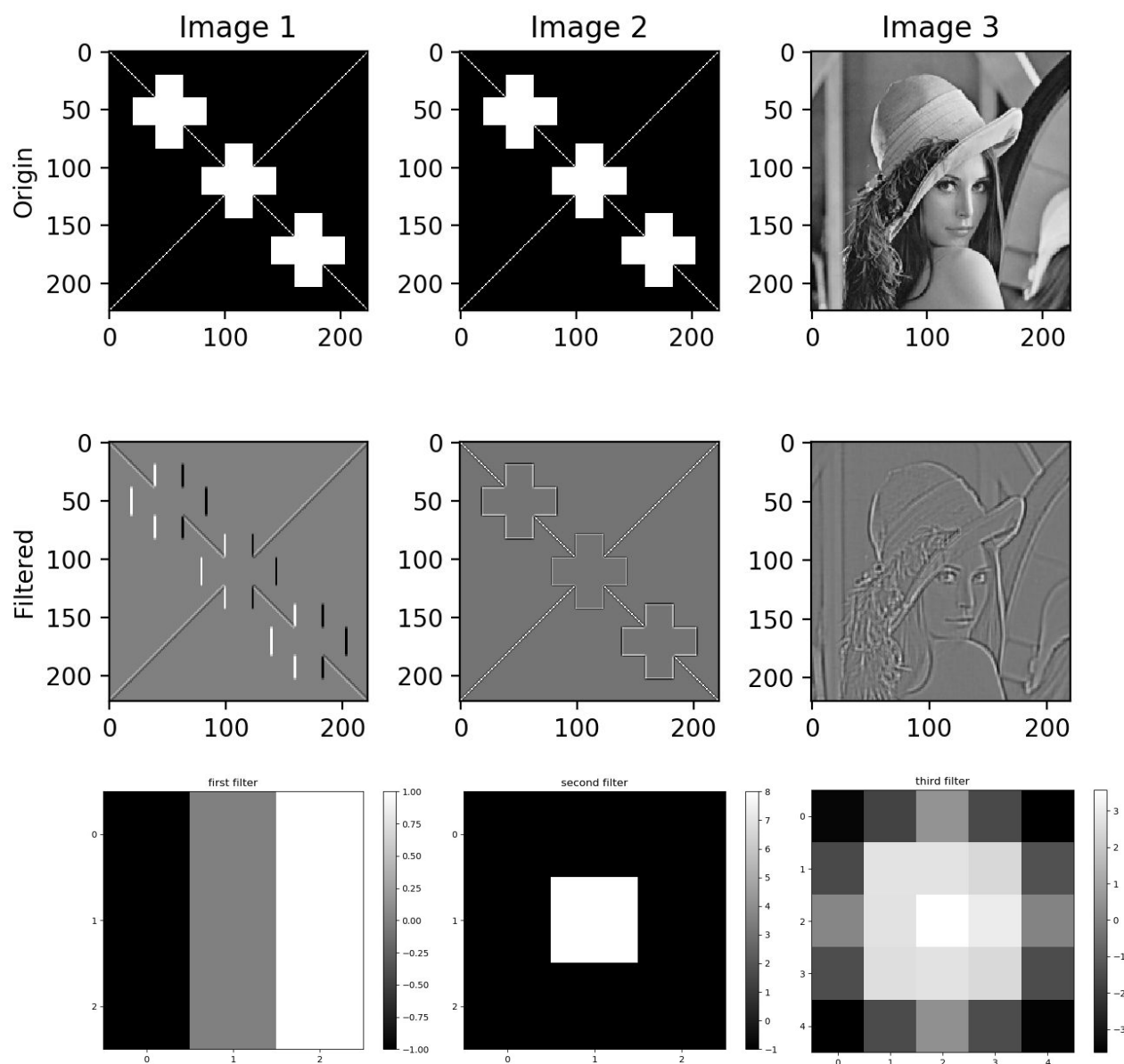
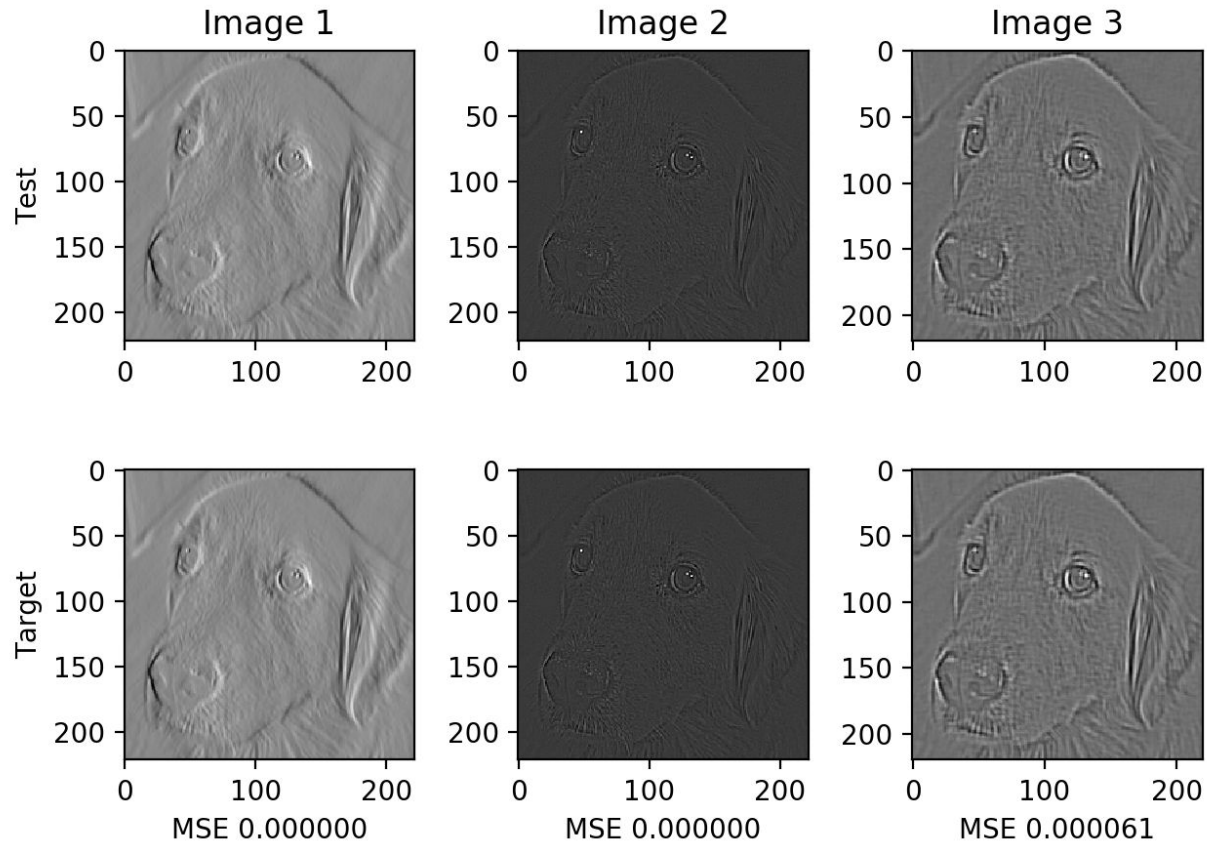


## 1. Training Result

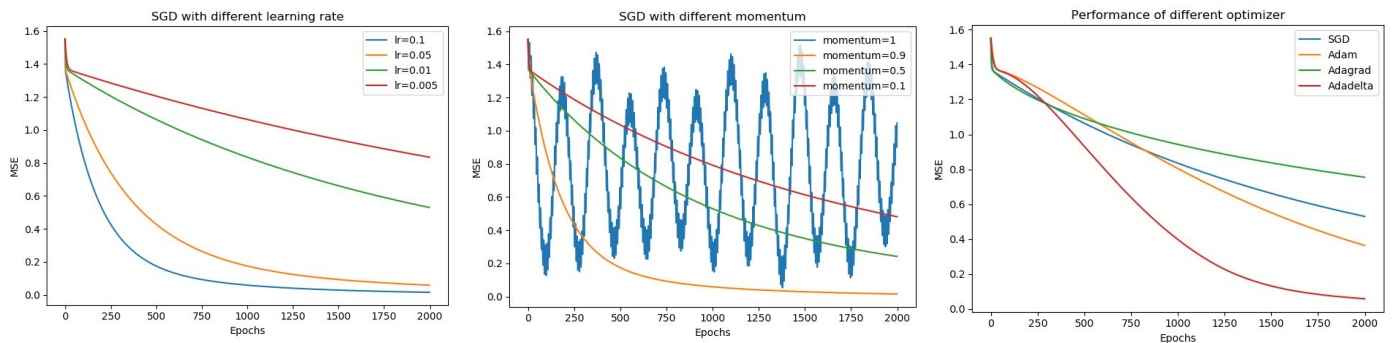
After we build the one layer 2D convolution network according to the image size and filter size, we can start to fitting the network and save the filter, then applied the learned filter to the testing image to compare the prediction result and expected result by calculating the MSE.

The optimizer we used to train the network is SGD with learning rate 0.1 and decay 0.001 (gradually decrease the lr through training, avoiding cost diverge), and we trained for 5000 epochs. The best testing results for the first and second filter are almost identical to the expected filtered image, with  $MSE = 0.000000$  (6 digits accuracy), the filtered image of the third filter has a  $MSE=0.000061$  compared to the testing image.





## 2. Performance of different optimizer configuration



In this assignment, we compared the performance of different optimizer configuration. All evaluation is carried out on the MSE of the third filter, and the initial weights are generated by the same random seed. If not specified, we used the keras default optimizer configuration (e.g.  $lr = 0.01$  in SGD). First, we tried SGD with different learning rate and momentum. We observed that, the SGD optimizer with the biggest learning rate(0.1) provided the best performance. Thanks to the simplicity of the network, the cost function didn't diverge for a bigger learning rate at the end. As for momentum, a momentum smaller but closer to one will bring a better result, if it's equals to one, the cost function will diverge and oscillate. Among the different optimizers we applied, adadelta provides the best performance at the end, which will adapt the learning rate according to gradients in previous epochs.