

Assignment 2 Report

CZ4042 Neural Networks

Name: He Yuhao

Matric. Number: U1722945E

Name: Li Guanlong

Matric. Number: U1722033H

**Introduction**

This project aims to apply Convolution Neural Network (CNN) and Recurrent Neural Network (RNN) to address real case classification problems. Different hyper-parameters and different ways of learning are applied, and by analysing the training costs as well as test accuracies, the differences among the models are discussed. In part A, CNN is utilized for the object recognition problem, the effectiveness of different numbers of feature maps in the convolution layers are analysed. Besides, different ways of learning gradient descent (GD) learning, GD with momentum, RMSProp algorithm, Adam optimizer and GD with dropout are explored. In part B,

**Methods**

For both parts, the programming language used in this project is Python, the framework used is TensorFlow.

In Part A, the CNN is designed for image recognition problem. Two convolutional layers with two max pooling layers are implemented. The first convolution layer has a window size of 9×9, and implemented using VALID padding and ReLU neurons, while the second convolution layer consists of a window size of 5×5, VALID padding and ReLU neurons. After each convolutional layer, there is a max pooling layer which has a pooling window of size 2×2, with stride of 2 and VALID padding. The second max pooling layer is connected to a fully connected layer of size 300 which also implements ReLU. Then, a softmax layer of size 10 serves as the output layer which will produce the predicted class of the object.

**Experiments and Results**

**Part A:** **Object Recognition**

1. After configuring the CNN properly, GD learning with mini-batch with batch size of 128, the 10000 training samples are used to train the CNN.

a. The training cost and test accuracy against learning epochs is plotted, as shown in Figure 1 and Figure 2 respectively.

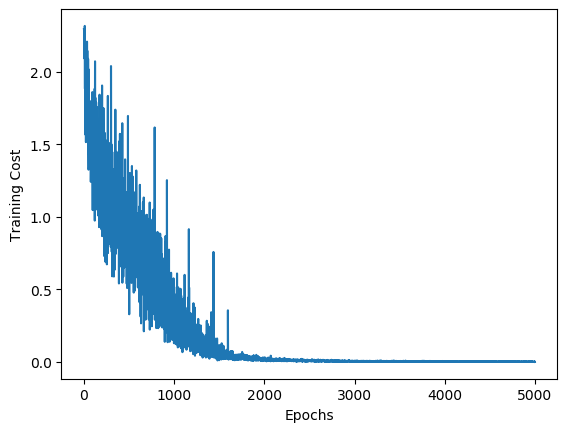
**** ****

Figure 1: Training cost Figure 2: Test accuracy

b. Two test images are randomly selected, the two images and their feature maps at both convolution layers and pooling layers are shown in Figure 3, 4, 5, 6, 7, 8, 9, 10, 11,12. Figure 3 is the first image, Figure 4, 5, 6, 7 are the first image’s feature maps at the first convolution layer, the first pooling layer, the second convolution layer and the second pooling layer respectively. Figure 8 is the second image, Figure 9, 10, 11, 12 are the second image’s feature maps at the first convolution layer, the first pooling layer, the second convolution layer and the second pooling layer respectively.

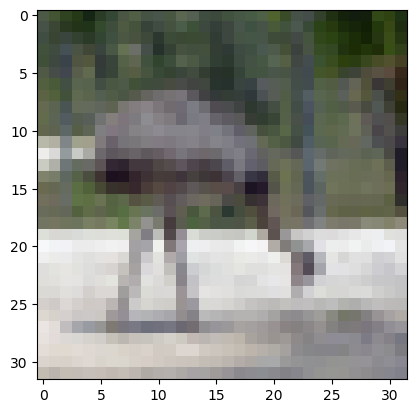


Figure 3: The first image

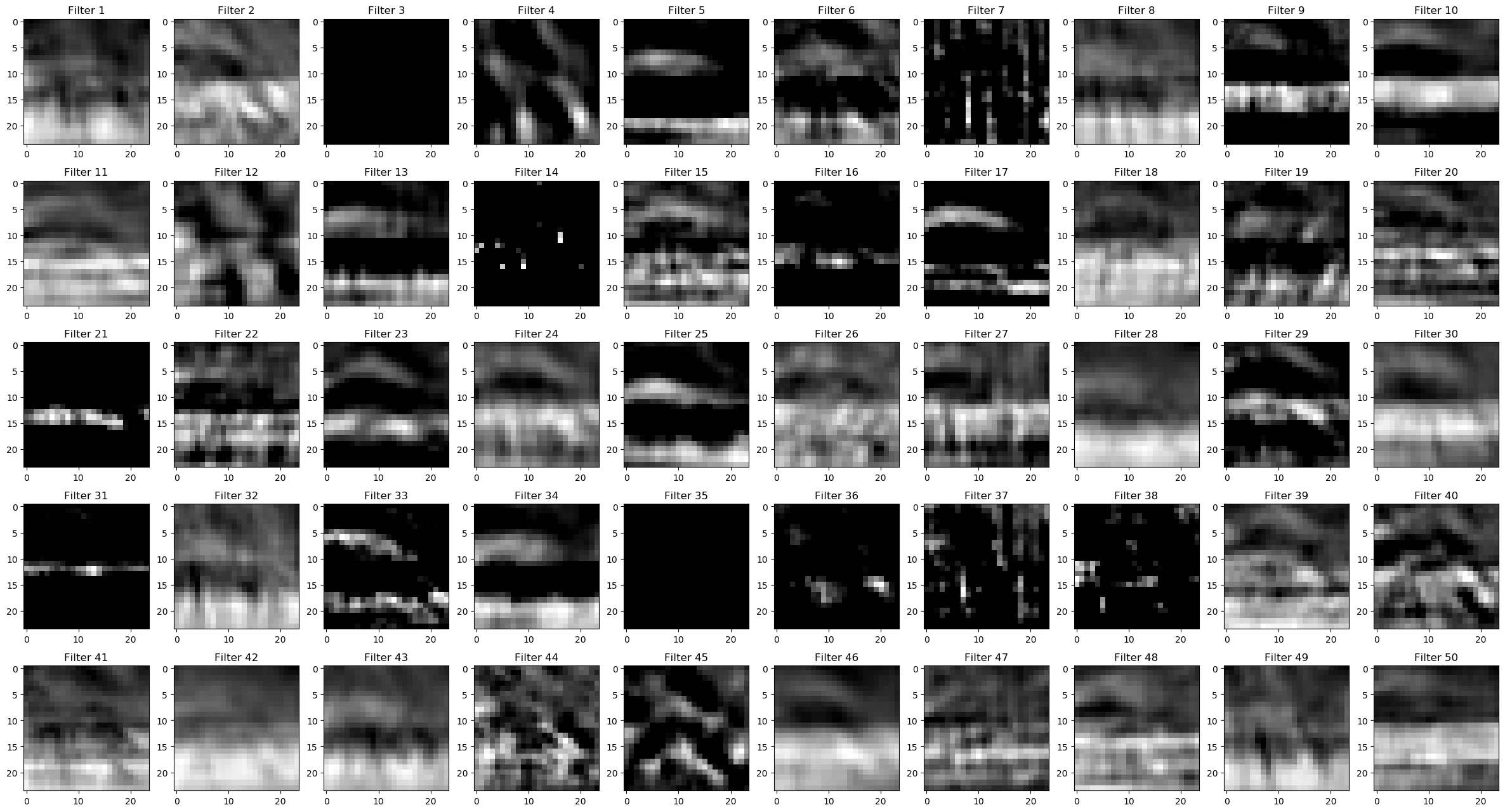


Figure 4: The first image’s feature maps at the first convolution layer

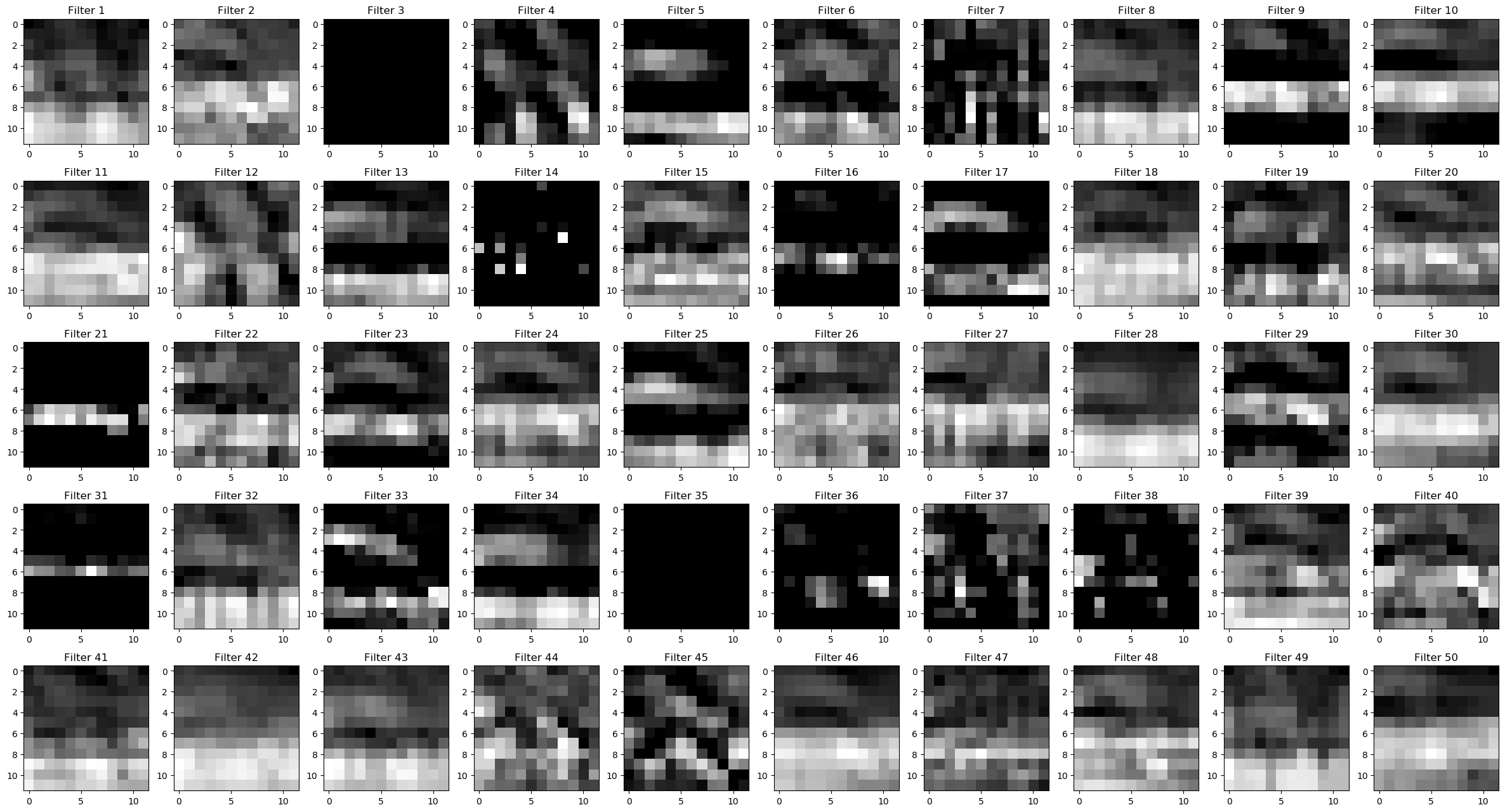


Figure 5: The first image’s feature maps at the first pooling layer

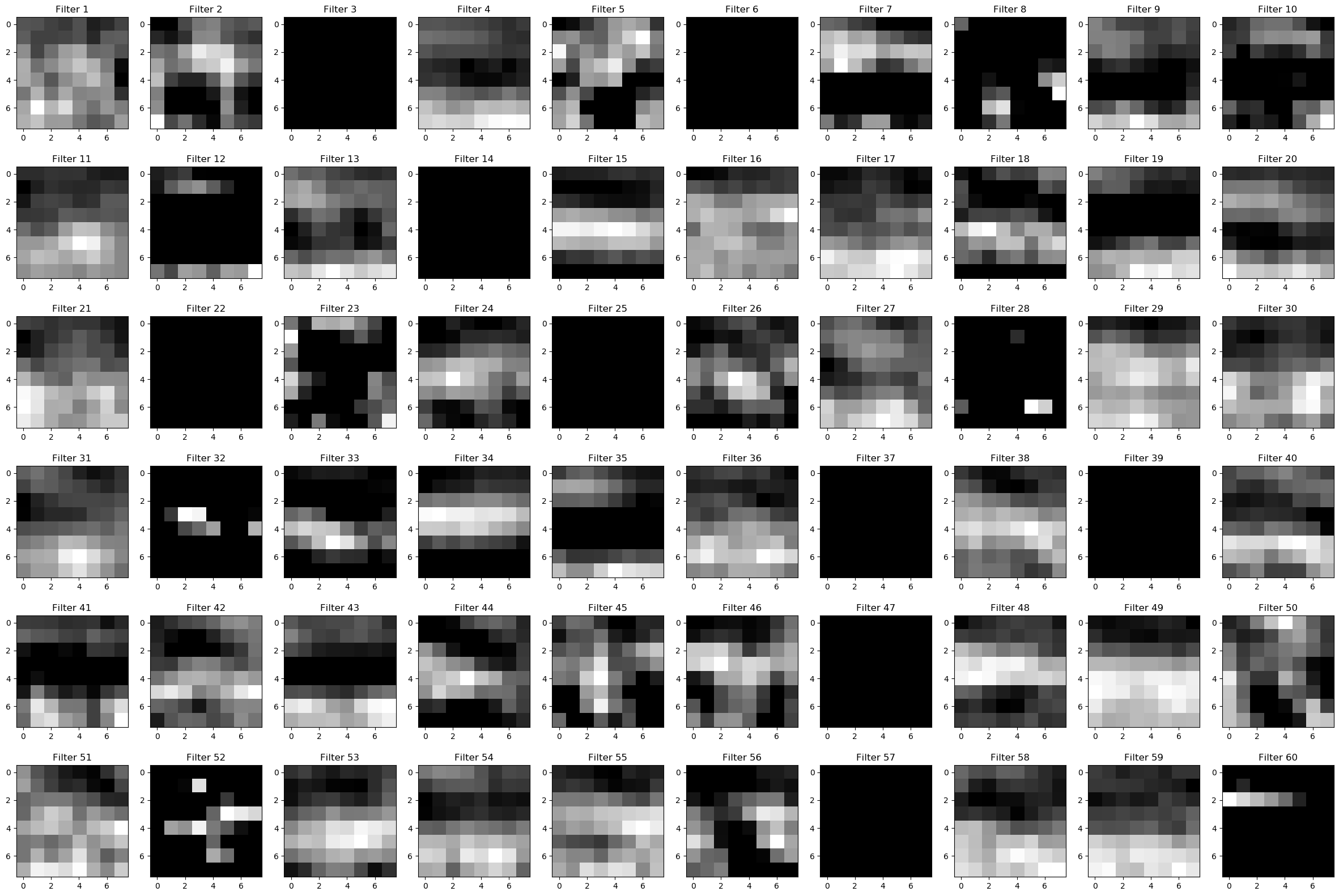


Figure 6: The first image’s feature maps at the second convolution layer

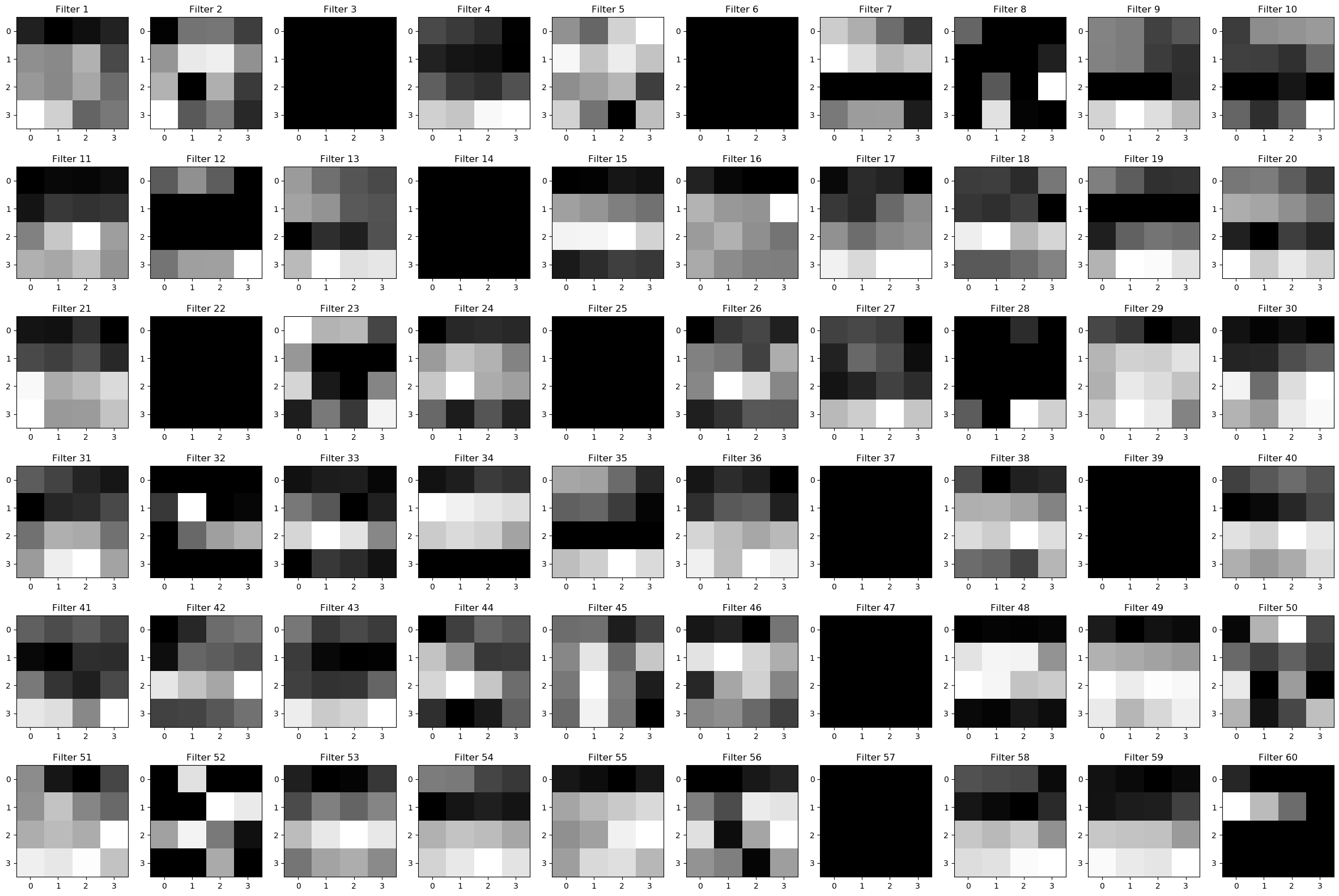


Figure 7: The first image’s feature maps at the second pooling layer

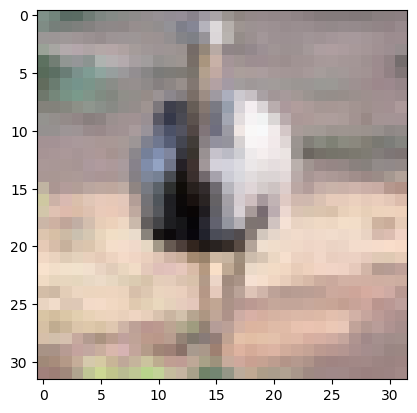


Figure 8: The second image

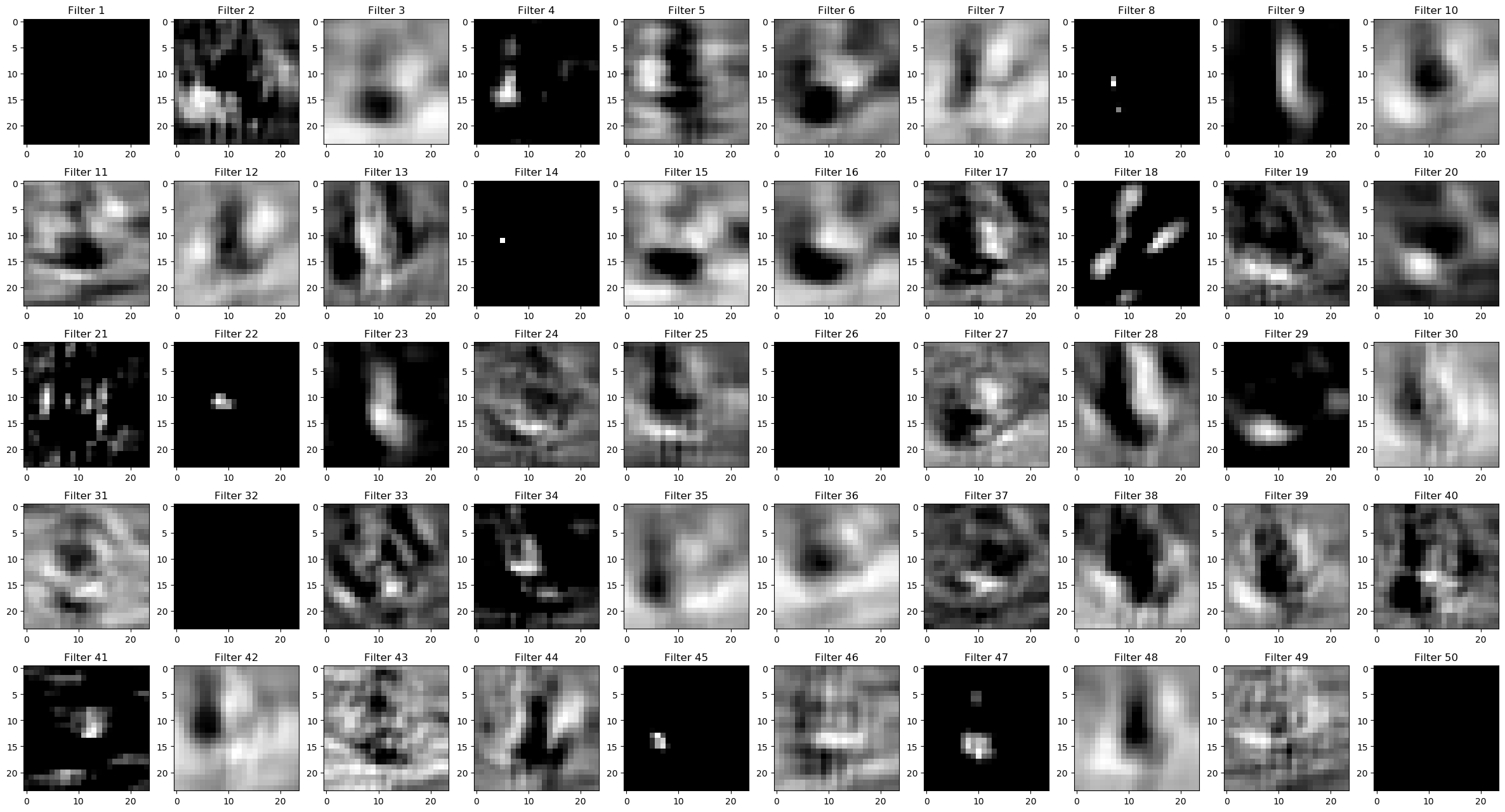


Figure 9: The second image’s feature maps at the first convolution layer

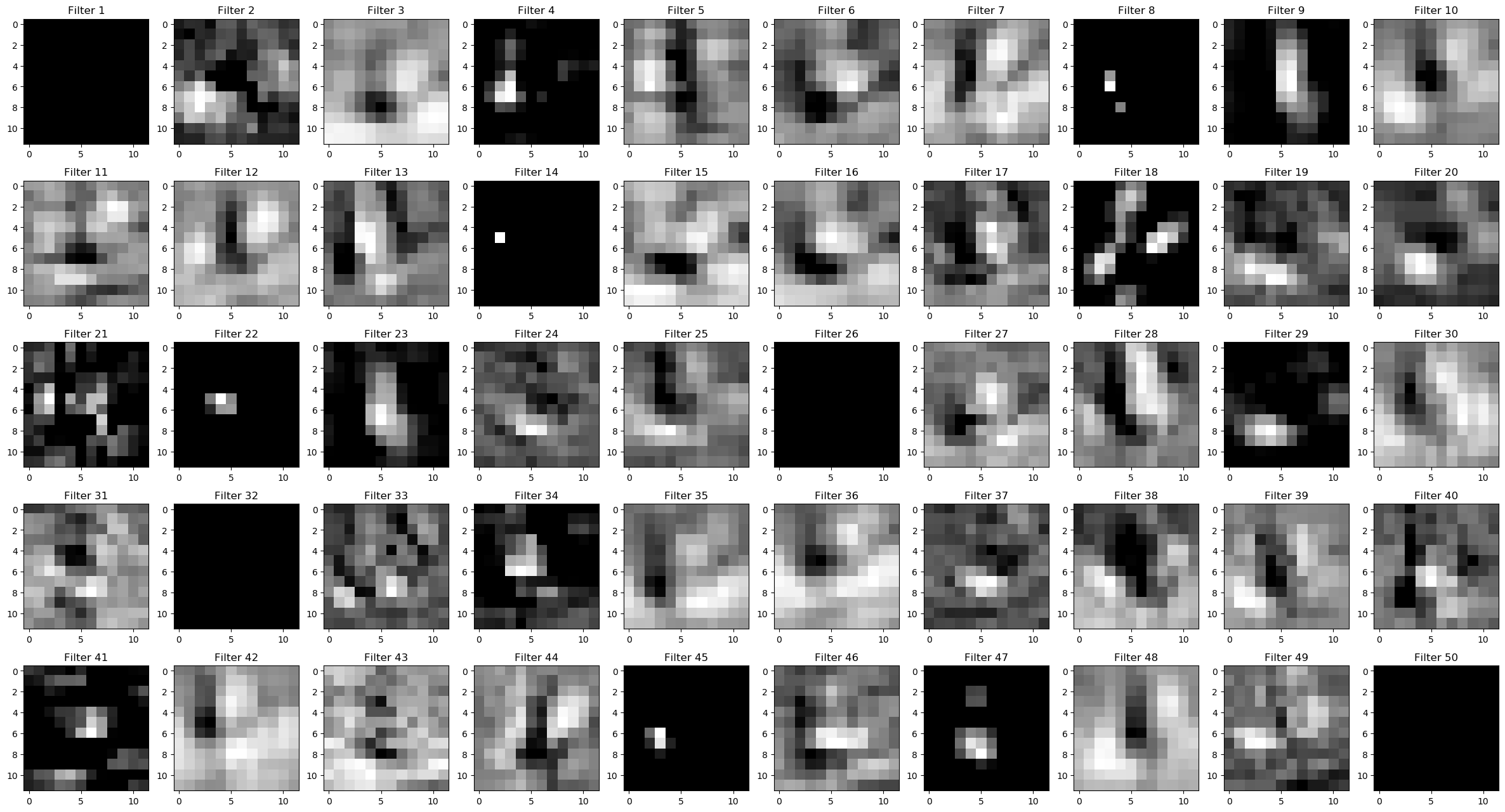


Figure 10: The second image’s feature maps at the first pooling layer

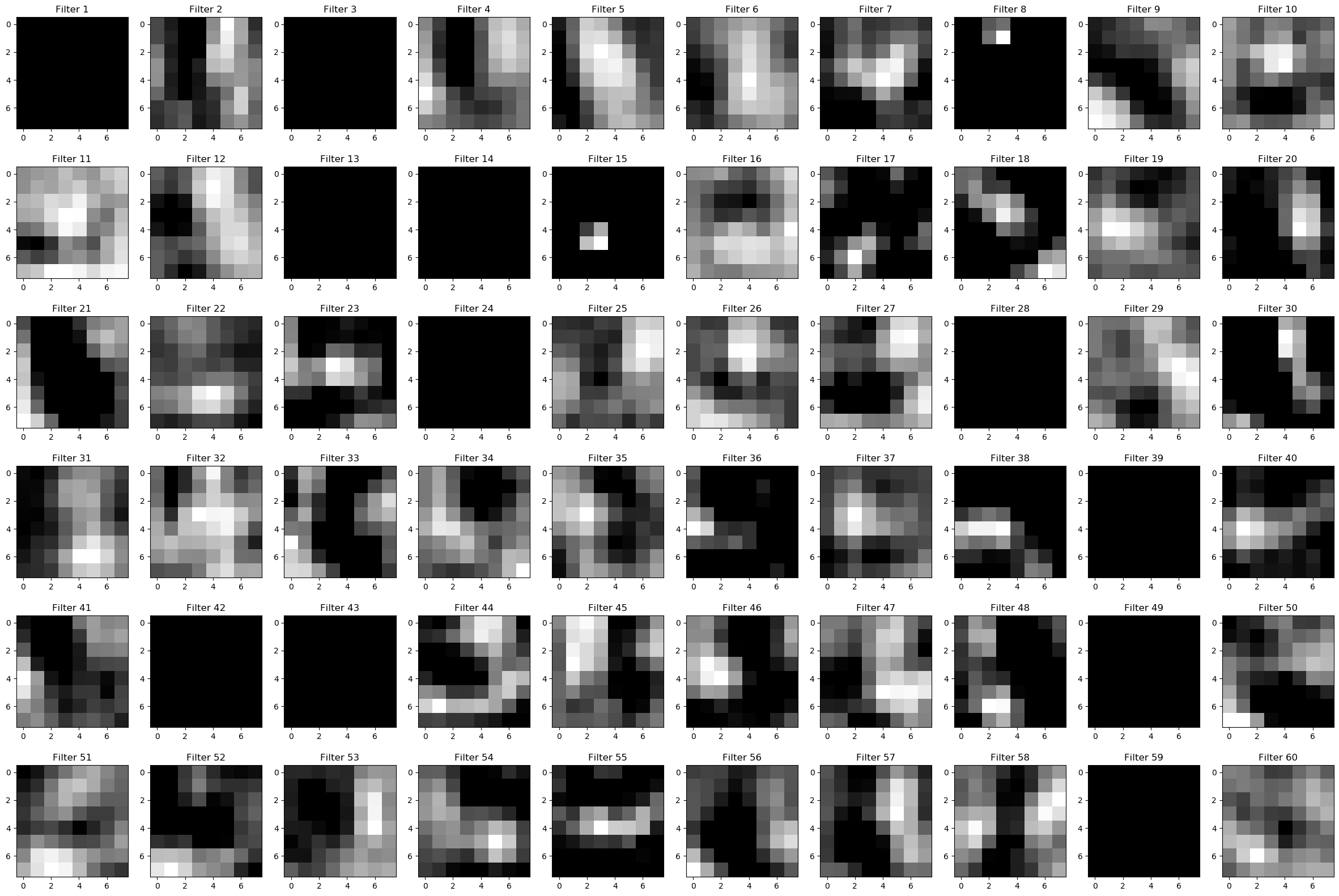


Figure 11: The second image’s feature maps at the second convolution layer

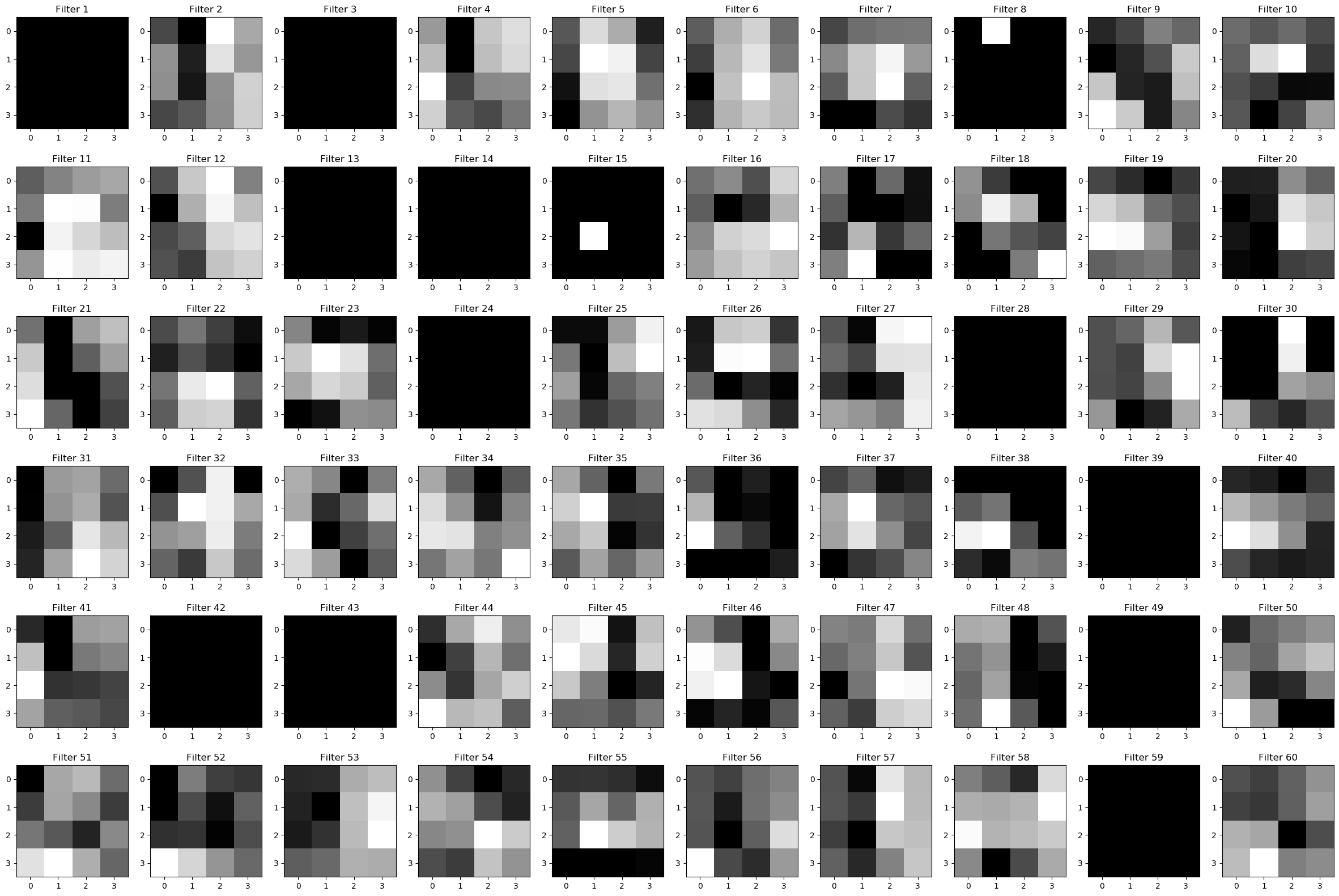


Figure 12: The second image’s feature maps at the second pooling layer

2. With a grid search, the [number of feature maps in first convolution layer, number of feature maps in second convolution layer] combinations of [40, 40], [40,50], [40, 60], [40, 70], [50, 40], [50,50], [50, 60], [50, 70], [60, 40], [60,50], [60, 60], [60, 70], [70, 40], [70,50], [70, 60], [70, 70] are examined one by one. The test accuracy against training epoch diagram for each of the combinations is shown in Figure 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28 respectively.

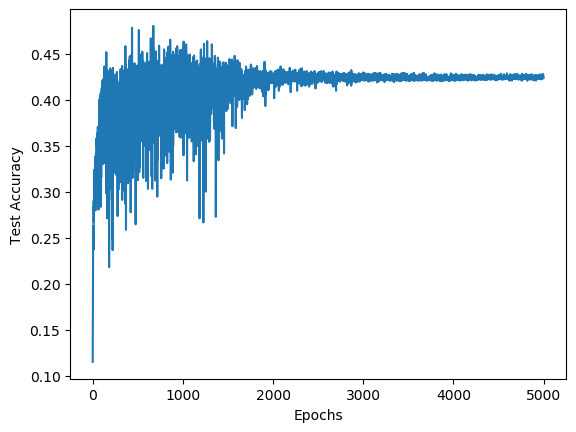
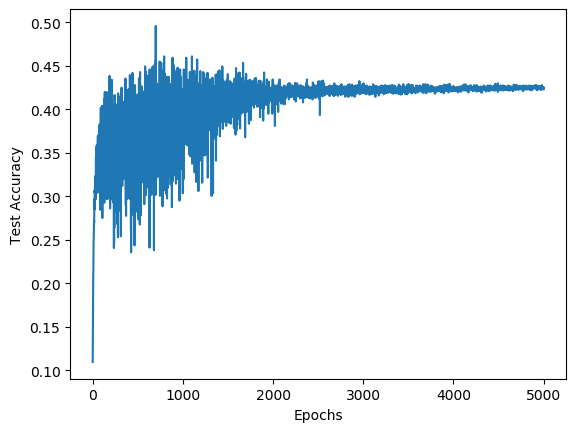
** **

Figure 13: Test accuracy for [40, 40] Figure 14: Test accuracy for [40, 50]

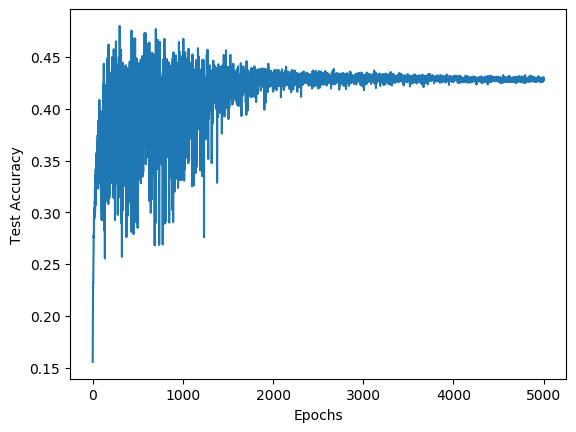
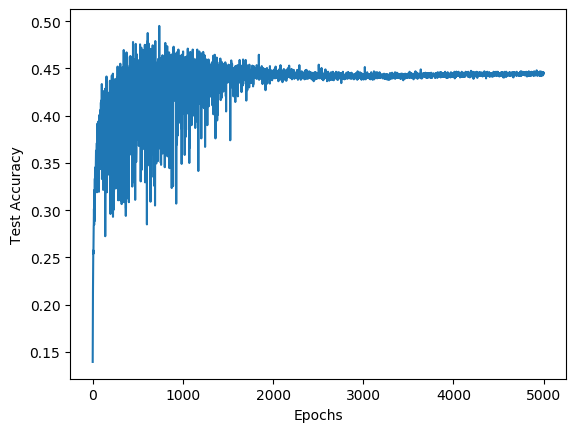
 

Figure 15: Test accuracy for [40, 60] Figure 16: Test accuracy for [40, 70]

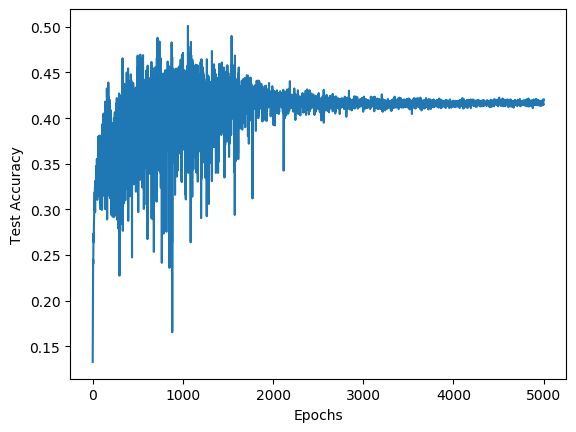
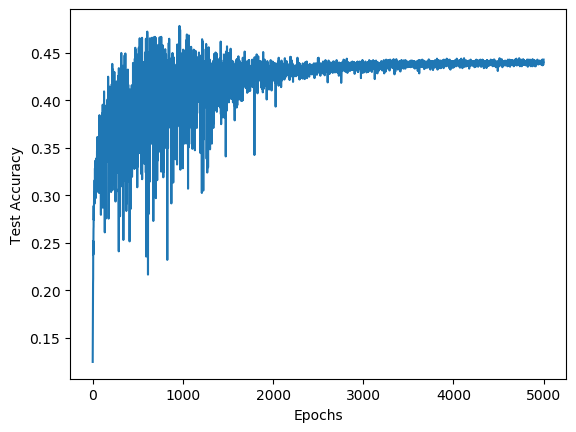
 

Figure 17: Test accuracy for [50, 40] Figure 18: Test accuracy for [50, 50]

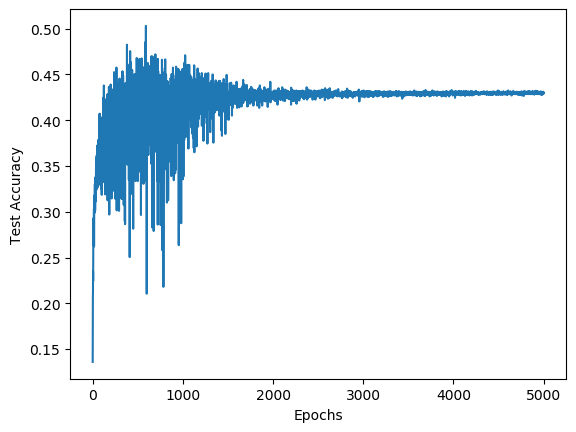
** **

Figure 19: Test accuracy for [50, 60] Figure 20: Test accuracy for [50, 70]

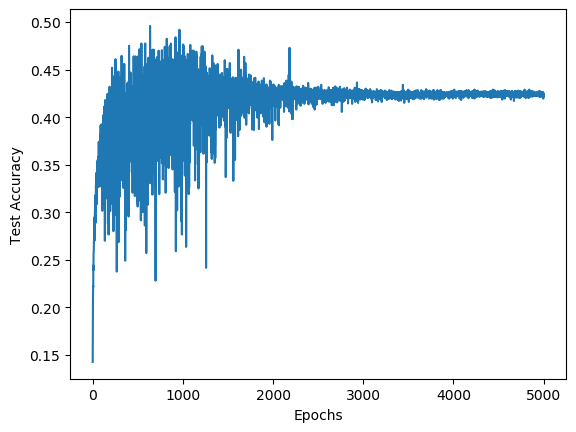
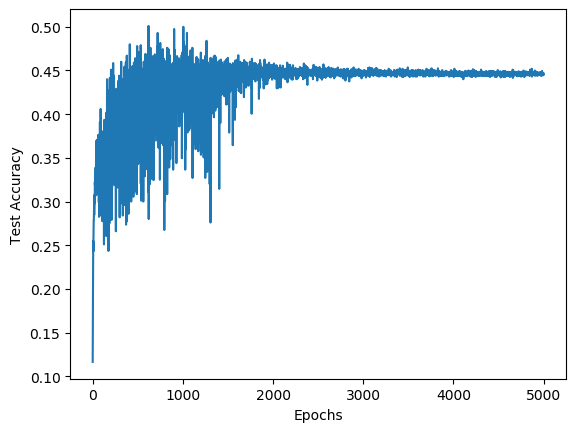
** **

Figure 21: Test accuracy for [60, 40] Figure 22: Test accuracy for [60, 50]

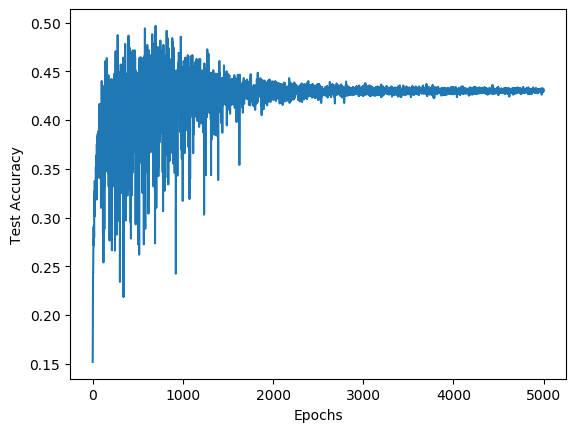
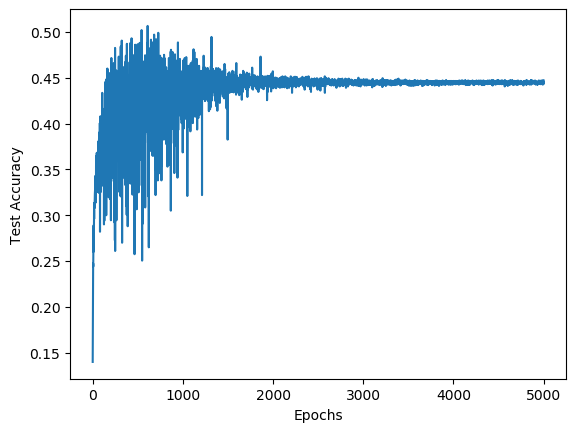
 ****

Figure 23: Test accuracy for [60, 60] Figure 24: Test accuracy for [60, 70]

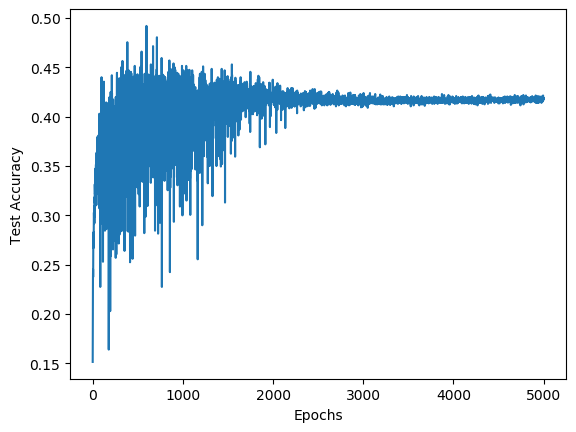
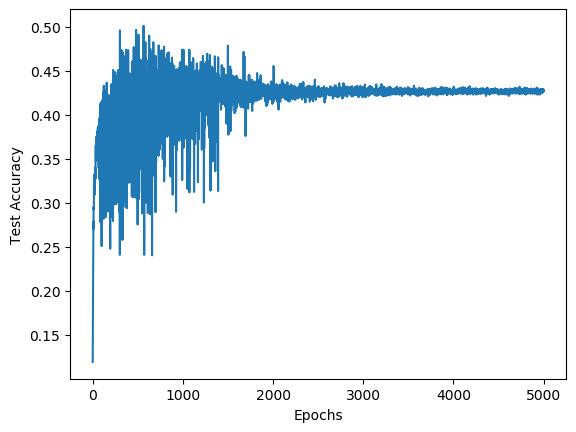
 

Figure 25: Test accuracy for [70, 40] Figure 26: Test accuracy for [70, 50]

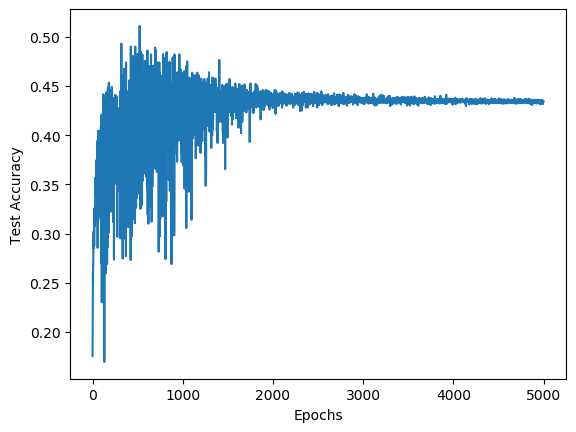
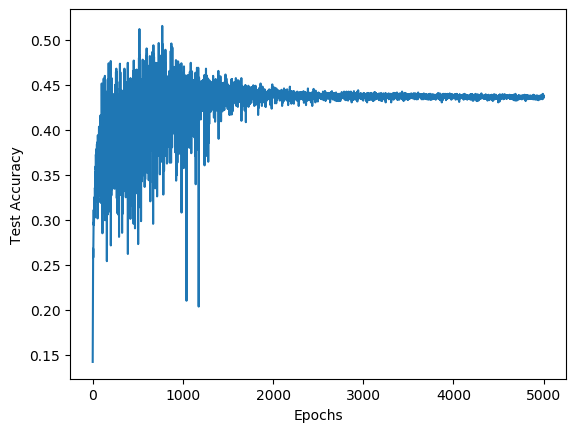
 

Figure 27: Test accuracy for [70, 60] Figure 28: Test accuracy for [70, 70]

From the figures above, and the data obtained by the grid search, it can be found that the combinations [40, 70], [60, 50] and [60, 70] can achieve the highest test accuracy, their test accuracies approximately converge to 0.445.

Since a smaller number of convolution layer feature maps will lead to a smaller usage of computational resources, the combination [40, 70] is selected to be the optimal number of feature maps. In other words, the optimal number of feature maps in the first convolution layer is 40, and the optimal number of feature maps in the second convolution layer is 70.

3. Using the optimal number of filters [40, 70], the CNN is trained by four different methods one at a time.

a. A momentum term 𝛾 = 0.1 is added to the network, with the momentum term, the training costs and test accuracies against epochs is plotted, as shown in Figure 29 and Figure 30.

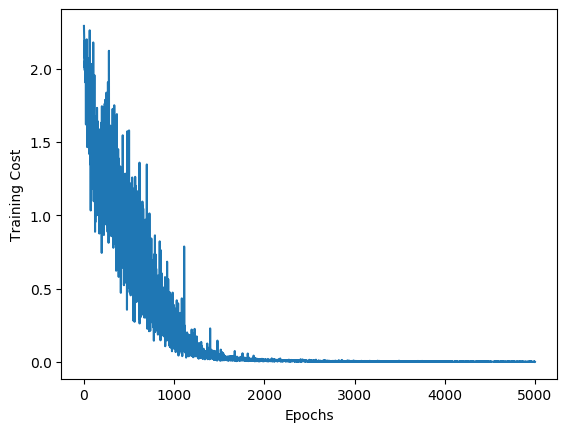
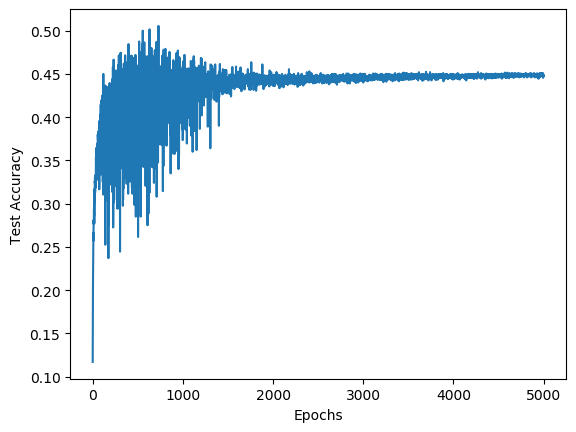
 

Figure 29: Training cost for momentum Figure 30: Test accuracy for momentum

b. When RMSProp algorithm is utilized for learning, the training costs and test accuracies against epochs is plotted, as shown in Figure 31 and Figure 32.

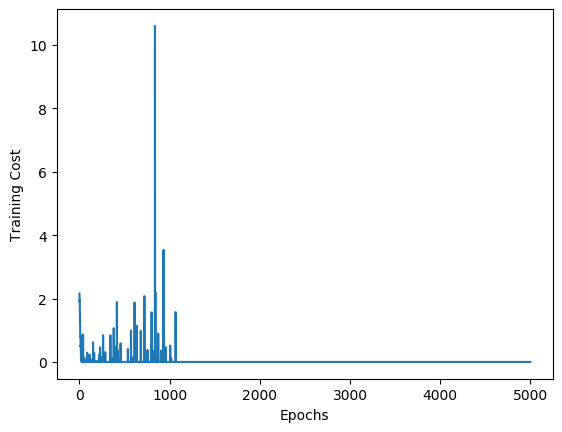
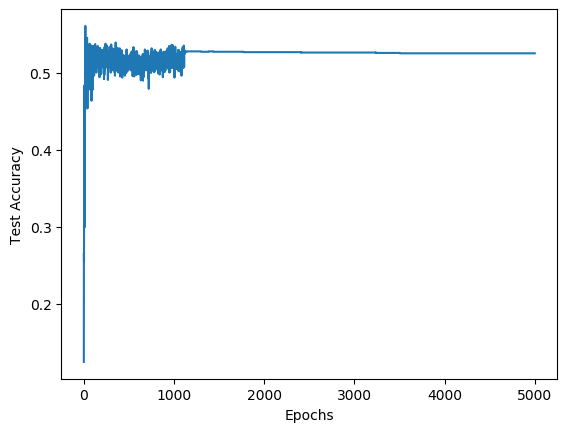
 

Figure 31: Training cost for RMSProp Figure 32: Test accuracy for RMSProp

c. When Adam optimizer is used for learning, the training costs and test accuracies against epochs is plotted, as shown in Figure 33 and Figure 34.

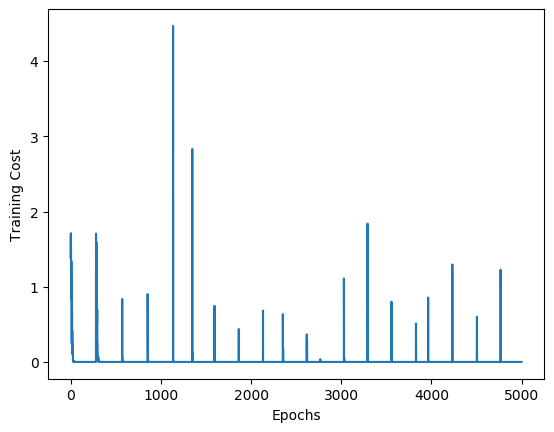
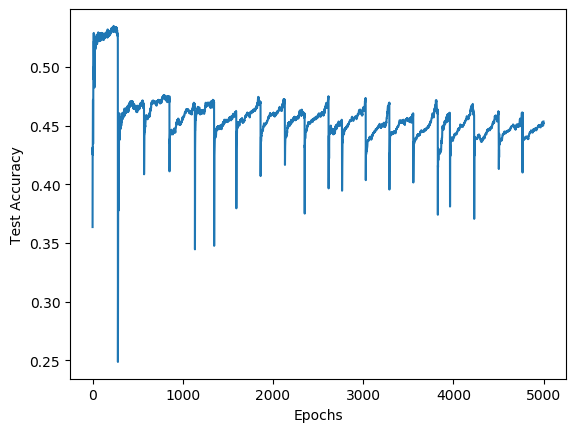
 

Figure 33: Training cost for Adam Figure 34: Test accuracy for Adam

d. When dropout with a rate of 0.5 is applied to the fully connected layer, the training costs and test accuracies against epochs is plotted, as shown in Figure 35 and Figure 36.

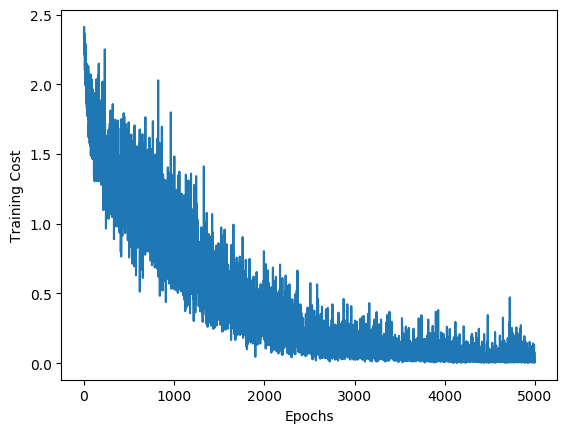
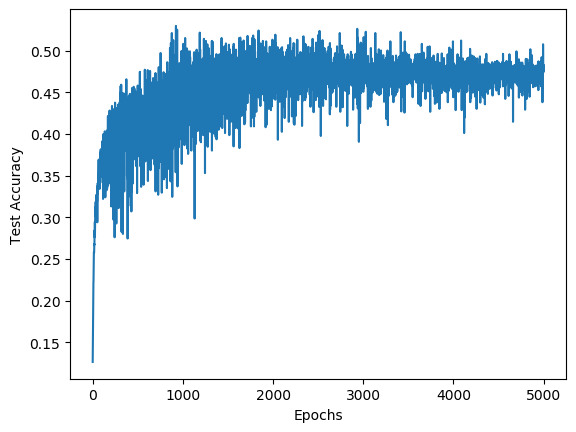
 

Figure 35: Training cost for dropout Figure 36: Test accuracy for dropout

4. From the accuracies of all the models from the previous parts, specifically Figure 16, 30, 32, 34, 46. It can be observed that:

* Among all the training methods, Adam optimizer and RMSProp algorithm can achieve the highest test accuracy, which approximately converges to 0.53. The test accuracies of GD with momentum and GD without any additional methods both converge to around 0.445. The test accuracies of GD with dropout has a fairly high tendency to fluctuate, by taking the average of the fluctuating accuracies, it can be concluded that the accuracies converge to around 0.47.
* Between Adam optimizer and RMSProp algorithm, it can be observed that Adam optimizer takes a shorter time to converge, but it has a periodic behaviour. That is to say, the test accuracy for the Adam optimizer can converge in less than 100 epochs, but if the training continues, the test accuracy will drop after around 200 epochs and rise to convergence again after less than 100 epochs. Therefore, from Figure 14, it can be concluded that for Adam optimizer, the training needs to be stopped after the accuracy converges for the first time, and this will yield a test accuracy of around 0.53. Meanwhile, it can be observed that for RMSProp algorithm, the test accuracy converges after around 1100 epochs, and tends to remain at around 0.53 even if the training continues. In general, both Adam optimizer and RMSProp algorithm can lead to a higher accuracy.
* In this particular CNN, the momentum does not have an obvious impact on the test accuracy and the epochs needed for the accuracy to converge.
* Adding a dropout to the fully connected layer is able to improve the test accuracy. However, it also makes the test accuracy prone to fluctuate.

**Conclusion**

In this project,