

## Problem 2: Incorporating CNNs

---

### Problem 2: Incorporating CNNs

[Training loss / accuracy curves for CNN training](#)

[Visualization of convolutional filters](#)

[Answers to inline questions about convolutional filters](#)

[Extra-Credit: Analysis on Trained Model](#)

---

## Training loss / accuracy curves for CNN training

Set hyperparameter:

```
optimizer = Adam(model.net, 1e-3)

batch_size = 128
epochs = 10
lr_decay = 0.98
lr_decay_every = 100
regularization = "l2"
reg_lambda = 0.001
```

Change Structure:

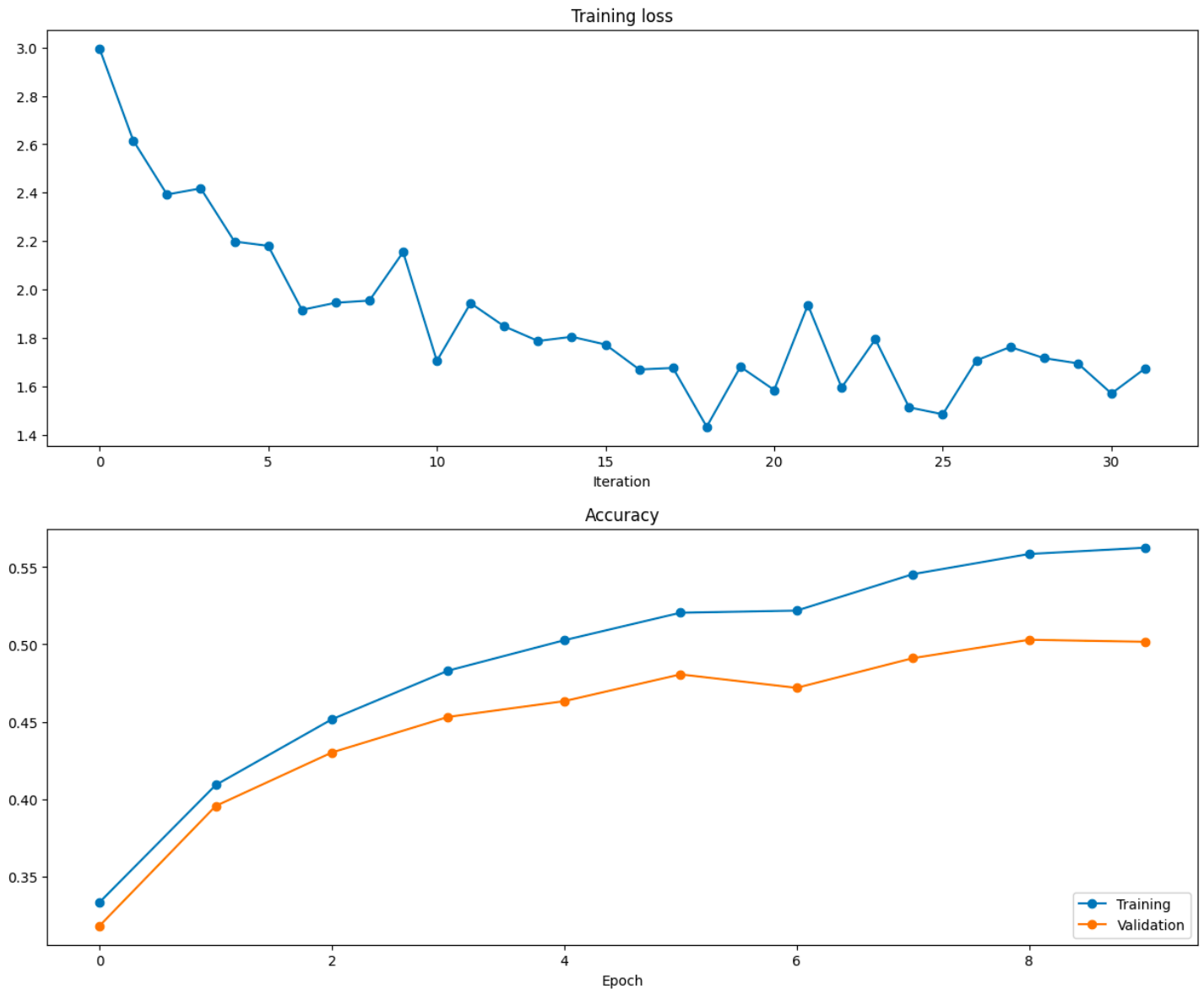
```
class SmallConvolutionalNetwork(Module):
    def __init__(self, keep_prob=0, dtype=np.float32, seed=None):
        self.seed=seed
        self.dropout = dropout
        self.net = sequential(
            ##### TODO: #####
            ConvLayer2D(3, 3, 32, 1, 0, name = "conv1"),
            gelu(name="lr1"),
            MaxPoolingLayer(3, 2, name = "pool1"),
            ConvLayer2D(32, 3, 32, 1, 0, name = "conv2"),
            gelu(name="lr2"),
            MaxPoolingLayer(3, 1, name = "pool2"),
            flatten(name = "flatten1"),
            fc(3200, 200, 0.02, name="fc1"),
            gelu(name="lr3"),
            dropout(keep_prob=0.6, seed=seed, name="dropout1"),
            fc(200, 20, 0.02, name="fc2"),
            ##### END #####
```

)

The accuracy meets the requirements:

Validation Accuracy: 50.17%

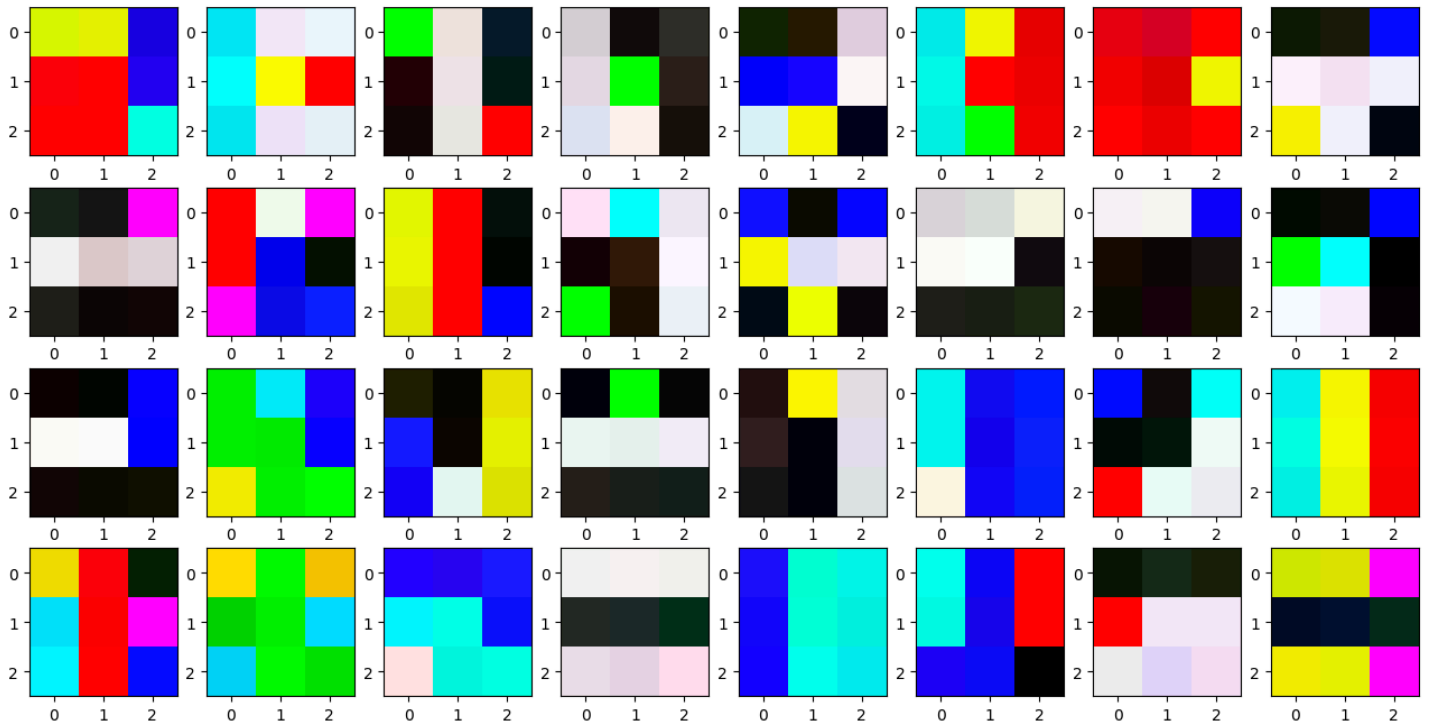
Testing Accuracy: 50.029999999999994%



## Visualization of convolutional filters

Visualize the first convolutional layer, which

$size = (3, 3, 3, 32) = (height, width, input\_channels, num\_filters(i.e. output\_channels))$



## Answers to inline questions about convolutional filters

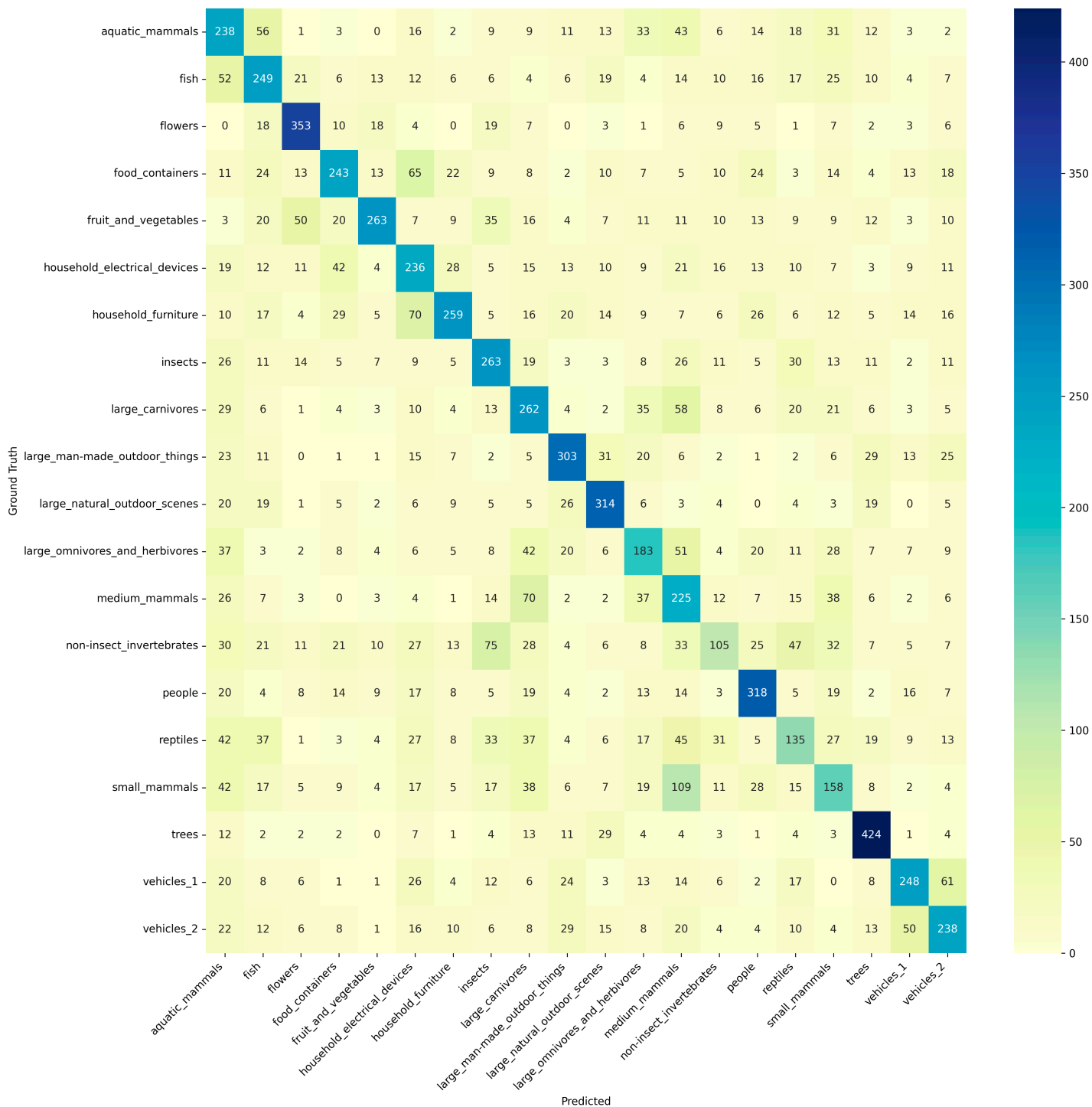
**Comment below on what kinds of filters you see. Include your response in your submission**

Most of the filters are like edge detection. In other words, they are learning different pattern. They're like our human's eyes, each of them represent a learner dealing with different image feature. Some filters focus on vertical feature, like the (3, 8) (count from top left to bottom right), some focus on horizontal feature, like the (1, 7), while others focus on diagonal feature, like (3, 2). Some detects the edge feature, like (3, 4).

Convolution layer can better focus on local information, which has similarity with human-beings. They learn and memorize different pattern through the convolutional filters. Thus, they can deal with new data by these filters.

## Extra-Credit: Analysis on Trained Model

I implement the confusion matrix function and plot the confusion matrix of my model's predictions on the test set by utilizing heat-map.



As shown in the heat-map figure, small mammals and medium mammals has the largest number of misclassification. They are both often confused by CNN.

There are also some other class like vehicles 1  $\Leftrightarrow$  vehicles 2, non-insect\_invertebrate  $\Leftrightarrow$  insects, medium mammals  $\Leftrightarrow$  large carnivores, household furniture  $\Leftrightarrow$  household electrical devices, ... , are the easily confused pair.

To sum up, these easily confused categories are also really close to our daily life. And the CNN, as one of the bionic intelligent system as human eye, also have something in common with human.