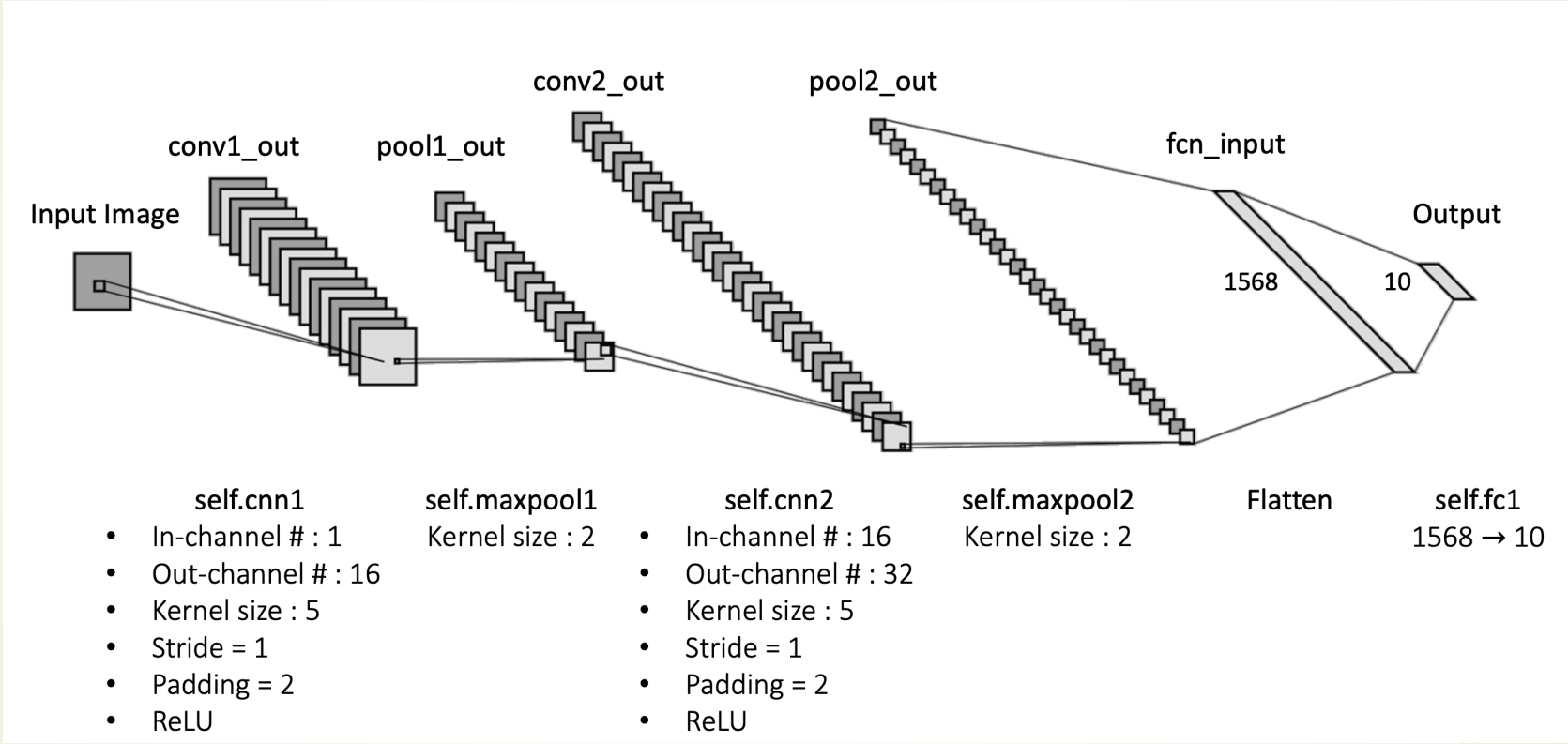
Implementation - Pytorch



Implementation - Pytorch

```
1  class CNNModel(torch.nn.Module):
2
3      def __init__(self):
4
5          super(CNNModel, self).__init__()
6
7          self.cnn1 = torch.nn.Conv2d(in_channels=1, out_channels=16,
8                                      kernel_size=5, stride=1, padding=2)
9
10         self.maxpool1 = torch.nn.MaxPool2d(kernel_size=2)
11
12         self.cnn2 = torch.nn.Conv2d(in_channels=16, out_channels=32,
13                                     kernel_size=5, stride=1, padding=2)
14
15         self.maxpool2 = torch.nn.MaxPool2d(kernel_size=2)
16
17         self.fc1 = torch.nn.Linear(32 * 7 * 7, 10)
```

Convolution
- In-channel # : 1
- Out-channel # : 16
- Kernel size : 5
- Stride = 1
- Padding = 2

Max Pool
Kernel size : 2

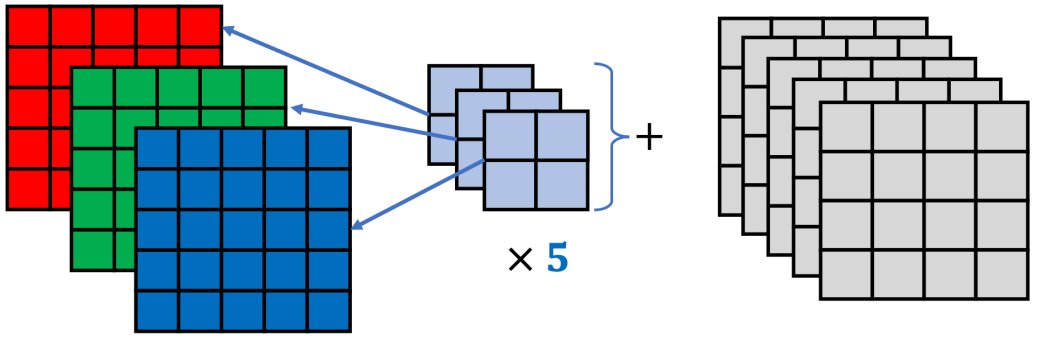
Convolution
- In-channel # : 16
- Out-channel # : 32
- Kernel size : 5
- Stride = 1
- Padding = 2

Max Pool
Kernel size : 2

FCN Layer
1568 → 10 neurons

Implementation - Pytorch

```
torch.nn.Conv2d(
    in_channels: 3
    out_channels: 5
    kernel_size: 2
)
```



Input
(1x3x5x5)

Kernels
(5x3x2x2)

Output
(1x5x4x4)

(N x Channels x Height x Width)

Implementation - Pytorch