Assignment #5:

Neural Network Architecture Design & Performance Evaluation

Objective: Demonstrate comprehension of architectural distinctions in neural networks and their effects on model performance.

Submission Deadline: Thursday, May 15, 2025, at 11:59 PM.

Task: Update your work from Assignment #3 by incorporating the following additions:

1. **Model Development Requirements:**

a) Construct and train the neural network architecture specified in the provided materials. Adjust the input dimension of the first nn.Linear() layer (initially set to 16x5x5) to match the output dimensions produced by the second nn.Conv2d() layer **when applied to your sign dataset**.

```
class Net(nn.Module):

    def __init__(self):
        super(Net, self).__init__()
        # 1 input image channel, 6 output channels, 5x5 square convolutio
        # kernel
        self.conv1 = nn.Conv2d(1, 6, 5)
        self.conv2 = nn.Conv2d(6, 16, 5)
        # an affine operation: y = Wx + b
        self.fc1 = nn.Linear(16 * 5 * 5, 120)
        self.fc2 = nn.Linear(120, 84)
        self.fc3 = nn.Linear(84, 10)

    def forward(self, x):
        # Max pooling over a (2, 2) window
        x = F.max_pool2d(F.relu(self.conv1(x)), (2, 2))
        # If the size is a square you can only specify a single number
        x = F.max_pool2d(F.relu(self.conv2(x)), 2)
        x = x.view(-1, self.num_flat_features(x))
        x = F.relu(self.fc1(x))
        x = F.relu(self.fc2(x))
        x = self.fc3(x)
        return x

    def num_flat_features(self, x):
        size = x.size()[1:]  # all dimensions except the batch dimension
        num_features = 1
        for s in size:
            num_features *= s
        return num_features
```

- **b.** Design and implement a ResNet (Residual Network) architecture.

2. **Performance Evaluation:**
   Conduct a comparative analysis of the two models' performance (e.g., accuracy, training efficiency, loss metrics) and summarize your findings in a structured report.