In [1]:
```
import torch
import torchvision
import numpy as np
import matplotlib.pyplot as plt
import torch.nn as nn
import torch.nn.functional as F
from torchvision.datasets import CIFAR100
import torchvision.transforms as transforms
from torchvision.utils import make_grid
from torch.utils.data.dataloader import DataLoader
from torch.utils.data import random_split,ConcatDataset
from torchsummary import summary

import ResNet
import ResNet1
import data
import CNN
import evaluate
import train
```

```
Files already downloaded and verified
```

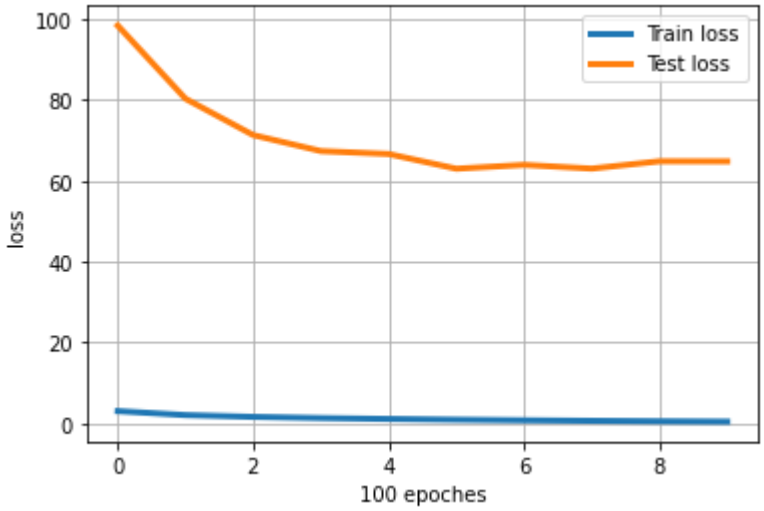
In [2]:
```
%%time
nets, train_loss_his, test_loss_his = train.train_smallNet(10000,1000,2)
```

```
iter:   1000, train loss: 2.99700932598114, test_loss: 98.45055413246155
accuracy: 37.44%
iter:   2000, train loss: 2.0108598576784136, test_loss: 80.339484333992
accuracy: 46.58%
iter:   3000, train loss: 1.5672681784629823, test_loss: 71.33919179439545
accuracy: 51.629999999999995%
iter:   4000, train loss: 1.2747611799240113, test_loss: 67.37476146221161
accuracy: 54.449999999999996%
iter:   5000, train loss: 1.0540951129198075, test_loss: 66.65164959430695
accuracy: 54.72%
iter:   6000, train loss: 0.8735138767957688, test_loss: 63.006292939186096
accuracy: 56.81%
iter:   7000, train loss: 0.7220853410661221, test_loss: 63.96186673641205
accuracy: 57.26%
iter:   8000, train loss: 0.5919309206902981, test_loss: 63.0322790145874
accuracy: 58.29%
iter:   9000, train loss: 0.4795257561802864, test_loss: 64.85145437717438
accuracy: 58.099999999999994%
iter:  10000, train loss: 0.389980237275362, test_loss: 64.8306782245636
accuracy: 59.08%
iter:   1000, train loss: 3.0127510035037997, test_loss: 97.2684006690979
accuracy: 37.480000000000004%
iter:   2000, train loss: 2.025610049843788, test_loss: 79.78214228153229
accuracy: 46.85%
iter:   3000, train loss: 1.5739695323705674, test_loss: 69.77957141399384
accuracy: 52.75%
iter:   4000, train loss: 1.27694296169281, test_loss: 65.25607240200043
accuracy: 55.50000000000001%
iter:   5000, train loss: 1.0506009578108788, test_loss: 62.81391906738281
accuracy: 57.14%
iter:   6000, train loss: 0.8662405729889869, test_loss: 62.89586067199707
accuracy: 57.24%
iter:   7000, train loss: 0.7148437358438968, test_loss: 61.62286639213562
accuracy: 58.660000000000004%
iter:   8000, train loss: 0.5846006797254085, test_loss: 62.152206897735596
accuracy: 59.3%
iter:   9000, train loss: 0.4741799736917019, test_loss: 63.83828115463257
```

```
accuracy: 59.019999999999996%
iter:  10000, train loss: 0.38358701016008856, test_loss: 65.69105410575867
accuracy: 58.830000000000005%
CPU times: user 12min 41s, sys: 3min 56s, total: 16min 38s
Wall time: 16min 42s
```

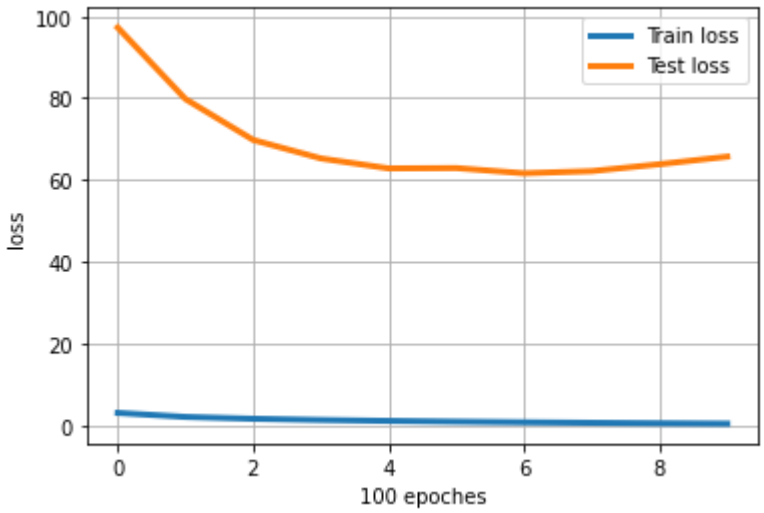
In [4]:
```
plt.plot(range(len(train_loss_his[0])),train_loss_his[0],'-',linewidth=3,label='Train l
plt.plot(range(len(train_loss_his[0])),test_loss_his[0],'-',linewidth=3,label='Test los
plt.xlabel('100 epoches')
plt.ylabel('loss')
plt.grid(True)
plt.legend()
```

Out[4]: <matplotlib.legend.Legend at 0x14d5e4c75070>



In [5]:
```
plt.plot(range(len(train_loss_his[1])),train_loss_his[1],'-',linewidth=3,label='Train l
plt.plot(range(len(train_loss_his[1])),test_loss_his[1],'-',linewidth=3,label='Test los
plt.xlabel('100 epoches')
plt.ylabel('loss')
plt.grid(True)
plt.legend()
```

Out[5]: <matplotlib.legend.Legend at 0x14d5db7f4340>



In [6]:
```
plt.plot(range(len(train_loss_his[2])),train_loss_his[2],'-',linewidth=3,label='Train l
plt.plot(range(len(train_loss_his[2])),test_loss_his[2],'-',linewidth=3,label='Test los
plt.xlabel('100 epoches')
```

```
plt.ylabel('loss')
plt.grid(True)
plt.legend()
```

In [ ]:
```
plt.plot(range(len(train_loss_his[3])),train_loss_his[3],'-',linewidth=3,label='Train l
plt.plot(range(len(train_loss_his[3])),test_loss_his[3],'-',linewidth=3,label='Test los
plt.xlabel('100 epoches')
plt.ylabel('loss')
plt.grid(True)
plt.legend()
```

In [ ]:
```
plt.plot(range(len(train_loss_his[4])),train_loss_his[4],'-',linewidth=3,label='Train l
plt.plot(range(len(train_loss_his[4])),test_loss_his[4],'-',linewidth=3,label='Test los
plt.xlabel('100 epoches')
plt.ylabel('loss')
plt.grid(True)
plt.legend()
```

In [8]:
```
trainDataLoader, testDataLoader =  data.loadData(250)
```

In [9]:
```
print(evaluate_2(nets,testDataLoader))
```

```
0.6328
```