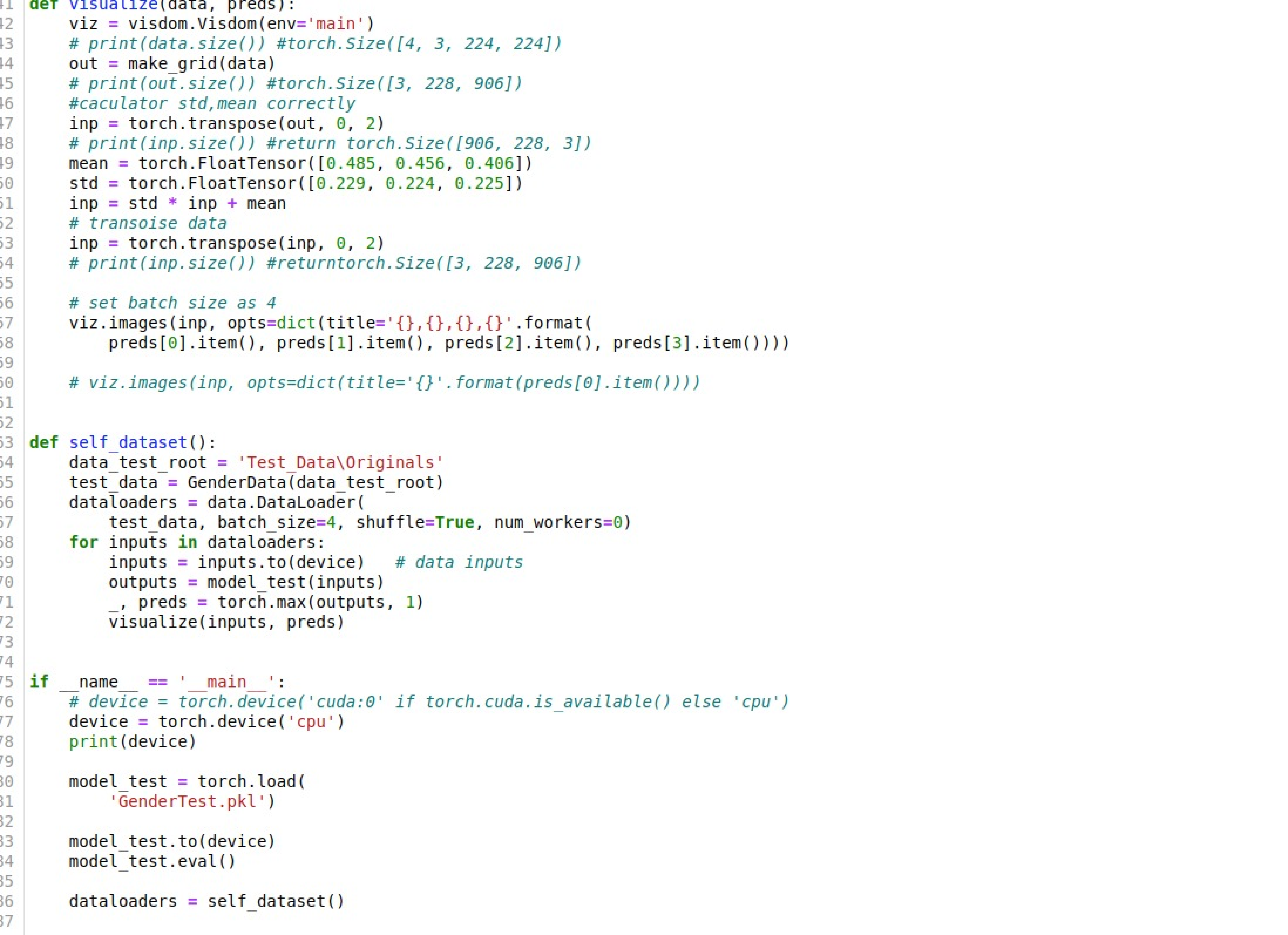

```
def visualize(data, preds):
    viz = visdom.Visdom(env='main')
    # print(data.size()) #torch.Size([4, 3, 224, 224])
    out = make_grid(data)
    # print(out.size()) #torch.Size([3, 228, 906])
    #caculator std,mean correctly
    inp = torch.transpose(out, 0, 2)
    # print(inp.size()) #return torch.Size([906, 228, 3])
    mean = torch.FloatTensor([0.485, 0.456, 0.406])
    std = torch.FloatTensor([0.229, 0.224, 0.225])
    inp = std * inp + mean
    # transoise data
    inp = torch.transpose(inp, 0, 2)
    # print(inp.size()) #returntorch.Size([3, 228, 906])

    # set batch size as 4
    viz.images(inp, opts=dict(title='{},{},{},{}'.format(
        preds[0].item(), preds[1].item(), preds[2].item(), preds[3].item())))

    # viz.images(inp, opts=dict(title='{}'.format(preds[0].item())))


def self_dataset():
    data_test_root = 'Test_Data\Originals'
    test_data = GenderData(data_test_root)
    dataloaders = data.DataLoader(
        test_data, batch_size=4, shuffle=True, num_workers=0)
    for inputs in dataloaders:
        inputs = inputs.to(device)   # data inputs
        outputs = model_test(inputs)
        _, preds = torch.max(outputs, 1)
        visualize(inputs, preds)


if __name__ == '__main__':
    # device = torch.device('cuda:0' if torch.cuda.is_available() else 'cpu')
    device = torch.device('cpu')
    print(device)

    model_test = torch.load(
        'GenderTest.pkl')

    model_test.to(device)
    model_test.eval()

    dataloaders = self_dataset()

```