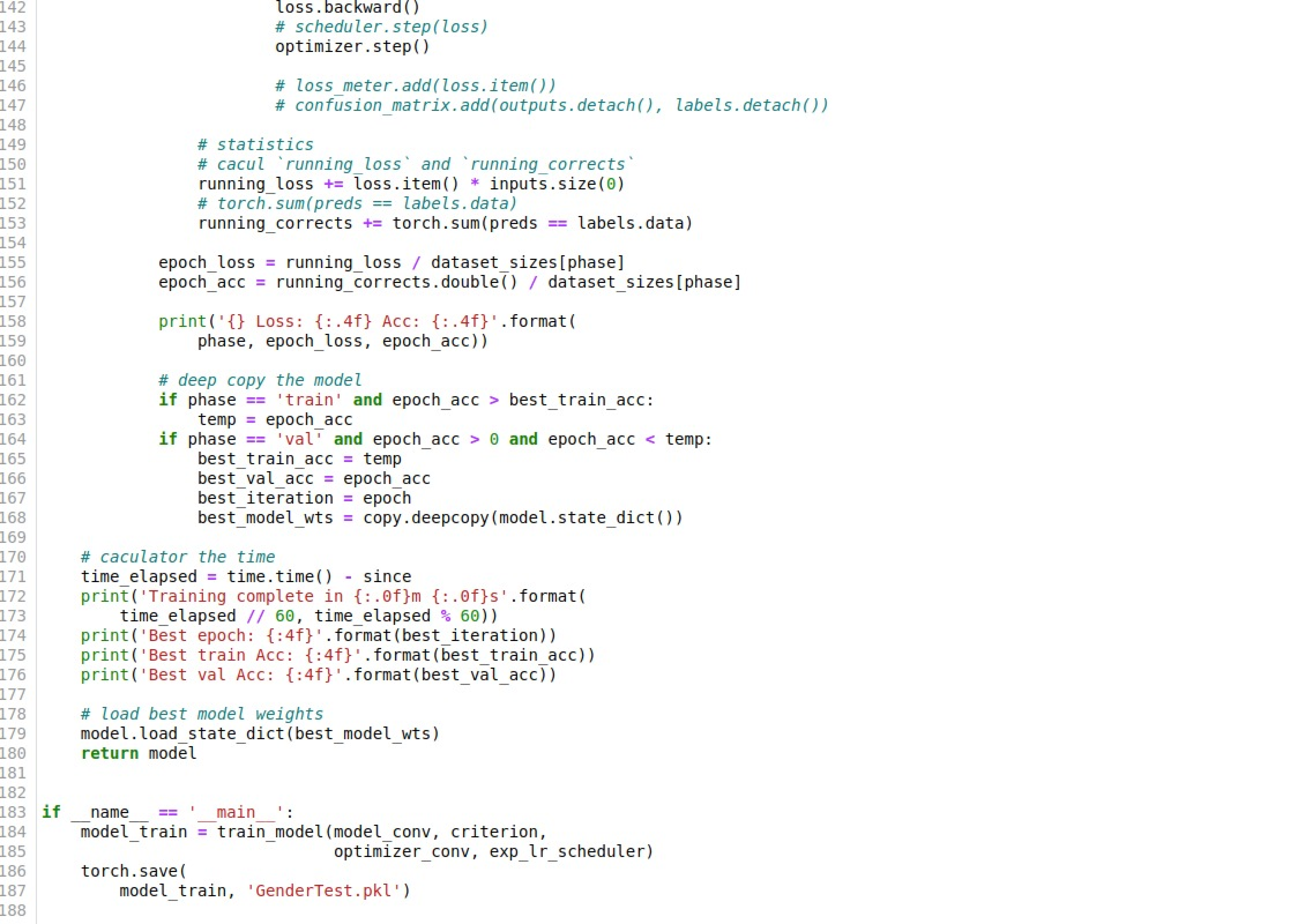

```python
                        loss.backward()
                        # scheduler.step(loss)
                        optimizer.step()

                        # loss_meter.add(loss.item())
                        # confusion_matrix.add(outputs.detach(), labels.detach())

                # statistics
                # cacul `running_loss` and `running_corrects`
                running_loss += loss.item() * inputs.size(0)
                # torch.sum(preds == labels.data)
                running_corrects += torch.sum(preds == labels.data)

            epoch_loss = running_loss / dataset_sizes[phase]
            epoch_acc = running_corrects.double() / dataset_sizes[phase]

            print('{} Loss: {:.4f} Acc: {:.4f}'.format(
                phase, epoch_loss, epoch_acc))

            # deep copy the model
            if phase == 'train' and epoch_acc > best_train_acc:
                temp = epoch_acc
            if phase == 'val' and epoch_acc > 0 and epoch_acc < temp:
                best_train_acc = temp
                best_val_acc = epoch_acc
                best_iteration = epoch
                best_model_wts = copy.deepcopy(model.state_dict())

    # caculator the time
    time_elapsed = time.time() - since
    print('Training complete in {:.0f}m {:.0f}s'.format(
        time_elapsed // 60, time_elapsed % 60))
    print('Best epoch: {:4f}'.format(best_iteration))
    print('Best train Acc: {:4f}'.format(best_train_acc))
    print('Best val Acc: {:4f}'.format(best_val_acc))

    # load best model weights
    model.load_state_dict(best_model_wts)
    return model


if __name__ == '__main__':
    model_train = train_model(model_conv, criterion,
                              optimizer_conv, exp_lr_scheduler)
    torch.save(
        model_train, 'GenderTest.pkl')
```