(1)

- Preprocess the x_test data by divided by 255 to normalize it.
- I use 4 convolution layers to build the model, which in order is 32, 64, 64, 128.
- The other hyperparameters are the same: kernel_size=(3, 3), padding=same, kernel_regularizer=l2(5e-4), activation=relu and add batch_normalization after each of the layer. I also add dropout(0.25) and add maxpooling(2, 2) to reduce overfitting and lower the spatial dimension.
- Batch_size is altered to 64 with 50 epochs and learning_rate=1e-3.
- Instead of using SGD, I use Adam as optimizer with beta_1(serve as momentum)=0.9, beta_2=0.999, epsilon=1e-4 and decay=learing_rate/epochs (which decreases learning_rate gradually).
- ImageDataGenerator is added as data augmentation (preprocessing).
- Train data is divided into 9:1 as training set and validation set.

Test Result:

```
[10] y_pred = model.predict(x_test)
     y_pred = np.argmax(y_pred, axis=1)

[ ] assert y_pred.shape == (10000,)

    (10000,)

[11] y_test = np.load("y_test.npy")
     print("Accuracy of my model on test set: ", accuracy_score(y_test, y_pred))

     Accuracy of my model on test set: 0.8337
```

(2) Used packages are attached in “requirements.txt”

(3) unzip the RAR file, then click on “inference.py”, then you can reproduce the result.