Exercise 2

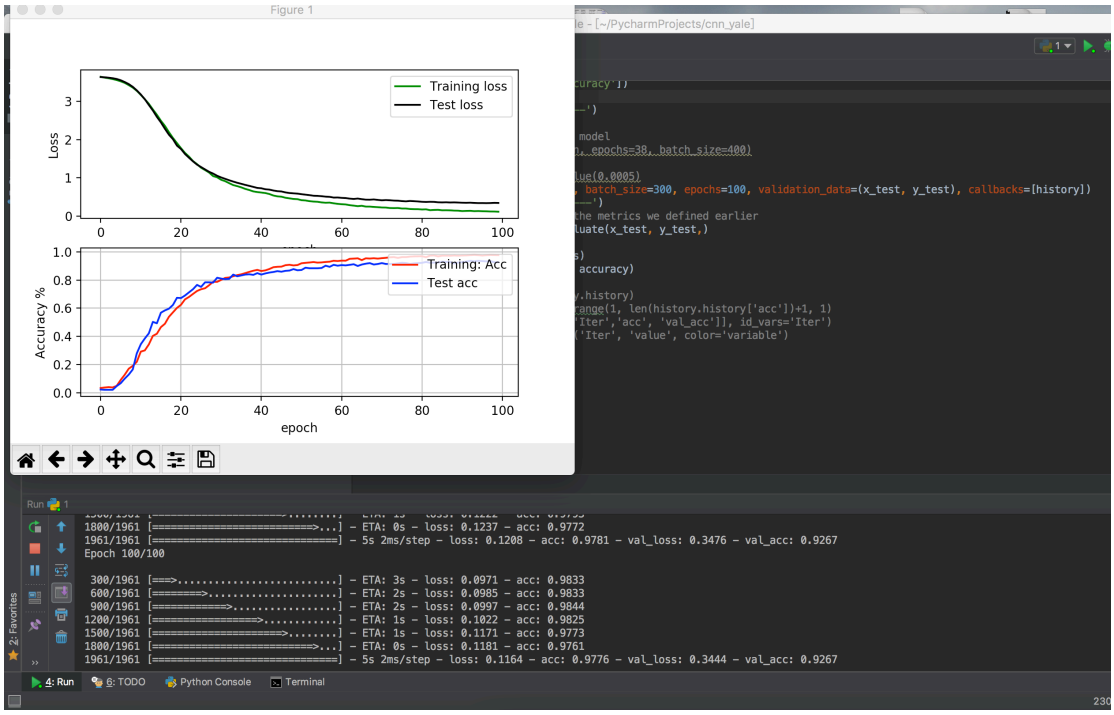
1.

```
# input image dimensions
img_rows, img_cols = 48, 48
# number of convolutional filters to use
nb_filters1 = 16
nb_filters2 = 36
# size of pooling area for max pooling
pool_size = (2, 2)
strides=(2, 2)
# convolution kernel size
kernel_size = (5, 5)
model=Sequential()
model.add(Convolution2D(nb_filters1, (kernel_size[0], kernel_size[1]),
                        padding='same',
                        input_shape=(48, 48, 1)))  # 卷积层1      output(16,48,48)
model.add(Activation('relu'))  # 激活层
model.add(MaxPooling2D(pool_size=pool_size, strides=strides, padding='same'))  # pool 1  output(16,24,24)
model.add(Convolution2D(nb_filters2, (kernel_size[0], kernel_size[1]), padding='same'))  # 卷积层2  output(36, 24, 24)
model.add(Activation('relu'))  # 激活层
model.add(MaxPooling2D(pool_size=pool_size, strides=strides, padding='same'))  # pool 2   output(36,12,12)
model.add(Dropout(0.5))  # 神经元随机失活
model.add(Flatten())  # 拉成一维数据  36*12*12=5184
model.add(Dense(512))  # 全连接层1
model.add(Activation('relu'))  # 激活层
model.add(Dense(nb_classes))  # 全连接层2
model.add(Activation('softmax'))  # Softmax评分
# Another way to define your optimizer
adam = Adam(lr=1e-4)

# We add metrics to get more results you want to see
model.compile(optimizer=adam,
              #loss='categorical_crossentropy',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 48, 48, 16)        416
_________________________________________________________________
activation_1 (Activation)    (None, 48, 48, 16)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 24, 24, 16)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 24, 24, 36)        14436
_________________________________________________________________
activation_2 (Activation)    (None, 24, 24, 36)        0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 12, 12, 36)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 12, 12, 36)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 5184)              0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               2654720
_________________________________________________________________
activation_3 (Activation)    (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 38)                19494
_________________________________________________________________
activation_4 (Activation)    (None, 38)                0
=================================================================
Total params: 2,689,066
Trainable params: 2,689,066
Non-trainable params: 0
```

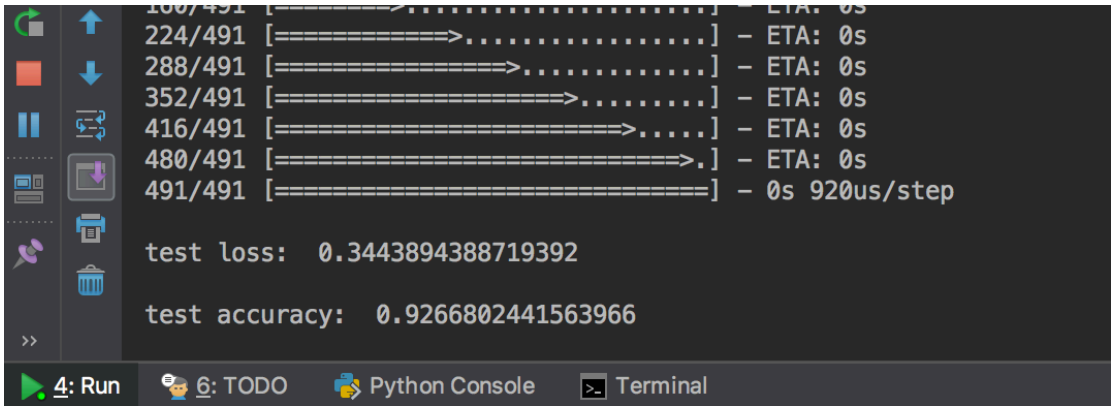
2.

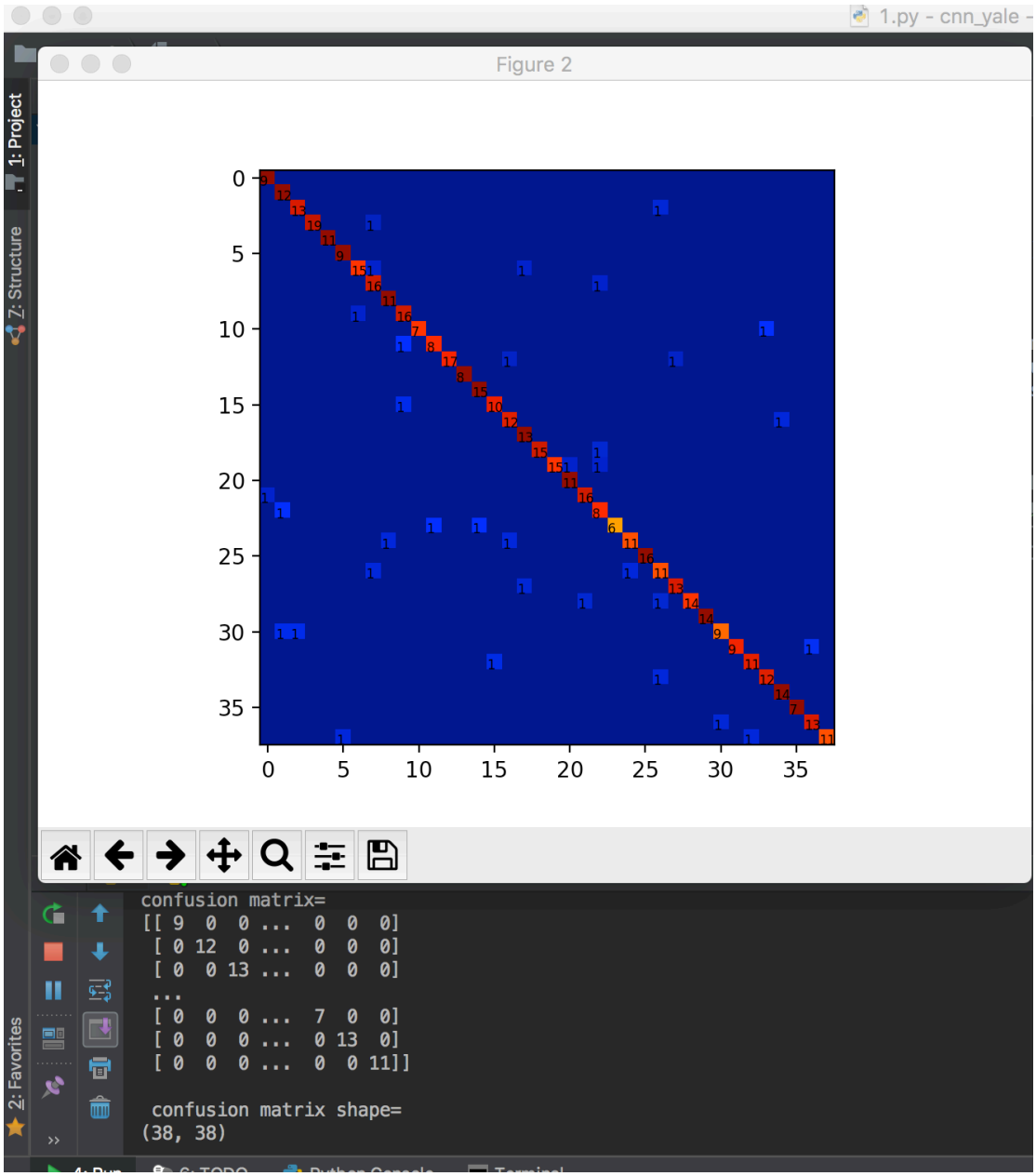


3.

```
model.fit(x_train, y_train, batch_size=300, epochs=100, validation_data=(x_test, y_test), callbacks=[history])
print('\nTesting ------------')
# Evaluate the model with the metrics we defined earlier
loss, accuracy = model.evaluate(x_test, y_test,)

print('\ntest loss: ', loss)
print('\ntest accuracy: ', accuracy)
```




1.py - cnn_yale -
Figure 2
1: Project
7: Structure
2: Favorites
confusion matrix=
[[9 0 0 ... 0 0 0]
 [0 12 0 ... 0 0 0]
 [0 0 13 ... 0 0 0]
 ...
 [0 0 0 ... 7 0 0]
 [0 0 0 ... 0 13 0]
 [0 0 0 ... 0 0 11]]
confusion matrix shape=
(38, 38)