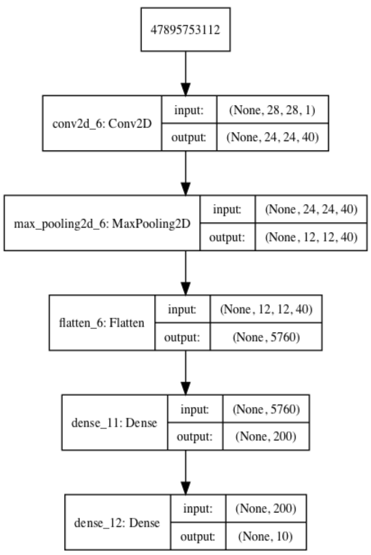
Task 2: Small CNN

```
cnn_model = models.Sequential()
cnn_model.add(Conv2D(40,
kernel_size=(5,5),activation='relu',input_shape=(img_rows,img_cols,1),padding='valid',strides=(1, 1)))
cnn_model.add(MaxPooling2D((2, 2)))
cnn_model.add(Flatten())
cnn_model.add(layers.Dense(200, activation = "relu"))
cnn_model.add(layers.Dense(10, activation = "softmax"))
cnn_model.summary()
```

```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 24, 24, 40)        1040
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 12, 12, 40)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 5760)              0
_________________________________________________________________
dense_1 (Dense)              (None, 200)               1152200
_________________________________________________________________
dense_2 (Dense)              (None, 10)                2010
=================================================================
Total params: 1,155,250
Trainable params: 1,155,250
Non-trainable params: 0
```



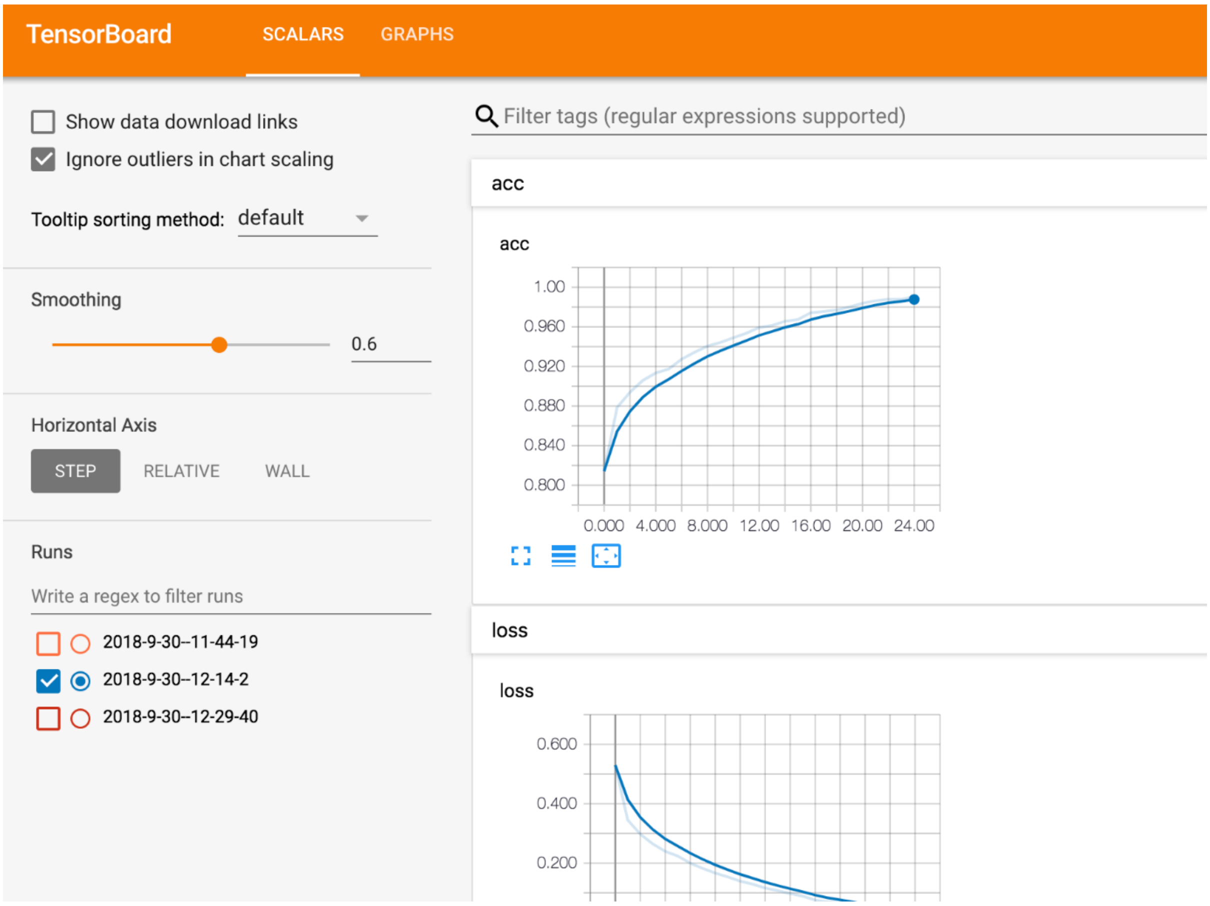

```
import time
t = time.localtime(time.time())
timeStamp = str(t.tm_year) + '-' + str(t.tm_mon) + '-' + str(t.tm_mday) + '--' + str(t.tm_hour) + '-'+str(t.tm_min) + '-
'+str(t.tm_sec)
tBoard = TensorBoard(log_dir='logs/{}'.format(timeStamp))
cnn_model.compile(
 optimizer = "adam",
 #loss = "categorical_crossentropy",
 loss = "sparse_categorical_crossentropy",
 metrics = ["accuracy"]
)
cnn_result = cnn_model.fit(train_x,target_train_y,
                batch_size=minibatch,
```

```
          epochs=epochs,
          validation_split = 0.2,
          shuffle=True,
          callbacks=[tBoard])
```

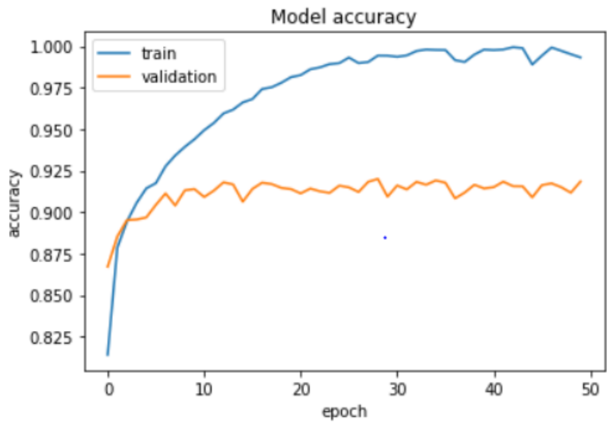
```
Train on 38400 samples, validate on 9600 samples
Epoch 1/50
38400/38400 [==============================] - 31s 807us/step - loss: 0
.5282 - acc: 0.8140 - val_loss: 0.3830 - val_acc: 0.8671
Epoch 2/50
38400/38400 [==============================] - 31s 795us/step - loss: 0
.3443 - acc: 0.8786 - val_loss: 0.3226 - val_acc: 0.8853
Epoch 3/50
38400/38400 [==============================] - 31s 795us/step - loss: 0
.2971 - acc: 0.8943 - val_loss: 0.2960 - val_acc: 0.8952
Epoch 4/50
38400/38400 [==============================] - 30s 776us/step - loss: 0
.2650 - acc: 0.9055 - val_loss: 0.2907 - val_acc: 0.8955
Epoch 5/50
38400/38400 [==============================] - 30s 776us/step - loss: 0
.2389 - acc: 0.9143 - val_loss: 0.2990 - val_acc: 0.8968
Epoch 6/50
38400/38400 [==============================] - 27s 705us/step - loss: 0
.2235 - acc: 0.9174 - val_loss: 0.2751 - val_acc: 0.9044
Epoch 7/50
38400/38400 [==============================] - 28s 727us/step - loss: 0
.1991 - acc: 0.9276 - val_loss: 0.2604 - val_acc: 0.9112
Epoch 8/50
38400/38400 [==============================] - 30s 769us/step - loss: 0
.1829 - acc: 0.9341 - val_loss: 0.2810 - val_acc: 0.9040
Epoch 9/50
38400/38400 [==============================] - 30s 785us/step - loss: 0
.1666 - acc: 0.9393 - val_loss: 0.2553 - val_acc: 0.9131
Epoch 10/50
38400/38400 [==============================] - 29s 768us/step - loss: 0
.1530 - acc: 0.9439 - val_loss: 0.2552 - val_acc: 0.9139
Epoch 11/50
38400/38400 [==============================] - 29s 746us/step - loss: 0
.1384 - acc: 0.9493 - val_loss: 0.2689 - val_acc: 0.9091
Epoch 12/50
38400/38400 [==============================] - 28s 721us/step - loss: 0
.1299 - acc: 0.9537 - val_loss: 0.2725 - val_acc: 0.9129
Epoch 13/50
38400/38400 [==============================] - 27s 710us/step - loss: 0
.1151 - acc: 0.9594 - val_loss: 0.2573 - val_acc: 0.9179
Epoch 14/50
38400/38400 [==============================] - 44s 1ms/step - loss: 0.1
074 - acc: 0.9617 - val_loss: 0.2655 - val_acc: 0.9167
Epoch 15/50
38400/38400 [==============================] - 50s 1ms/step - loss: 0.0
965 - acc: 0.9661 - val_loss: 0.3047 - val_acc: 0.9063
Epoch 16/50
38400/38400 [==============================] - 49s 1ms/step - loss: 0.0
895 - acc: 0.9681 - val_loss: 0.2876 - val_acc: 0.9140
Epoch 17/50
38400/38400 [==============================] - 50s 1ms/step - loss: 0.0
762 - acc: 0.9741 - val_loss: 0.2985 - val_acc: 0.9177
Epoch 18/50
```

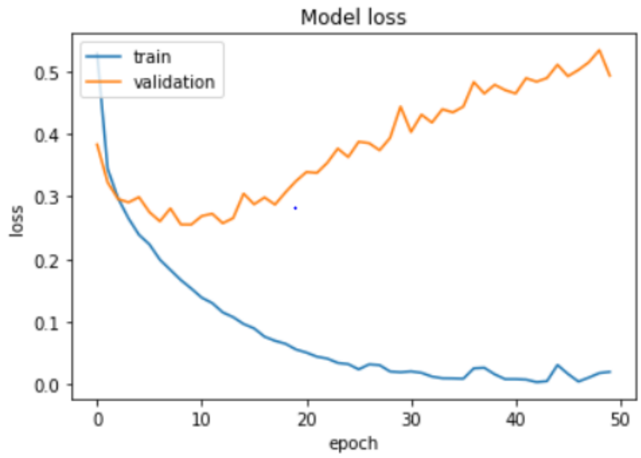
```
38400/38400 [==============================] - 50s 1ms/step - loss: 0.0
696 - acc: 0.9752 - val_loss: 0.2871 - val_acc: 0.9169
Epoch 19/50
38400/38400 [==============================] - 50s 1ms/step - loss: 0.0
649 - acc: 0.9779 - val_loss: 0.3067 - val_acc: 0.9146
Epoch 20/50
38400/38400 [==============================] - 50s 1ms/step - loss: 0.0
558 - acc: 0.9813 - val_loss: 0.3245 - val_acc: 0.9139
Epoch 21/50
38400/38400 [==============================] - 49s 1ms/step - loss: 0.0
508 - acc: 0.9825 - val_loss: 0.3392 - val_acc: 0.9113
Epoch 22/50
38400/38400 [==============================] - 50s 1ms/step - loss: 0.0
443 - acc: 0.9860 - val_loss: 0.3377 - val_acc: 0.9142
Epoch 23/50
38400/38400 [==============================] - 49s 1ms/step - loss: 0.0
412 - acc: 0.9872 - val_loss: 0.3539 - val_acc: 0.9125
Epoch 24/50
38400/38400 [==============================] - 49s 1ms/step - loss: 0.0
344 - acc: 0.9892 - val_loss: 0.3766 - val_acc: 0.9116
Epoch 25/50
38400/38400 [==============================] - 50s 1ms/step - loss: 0.0
324 - acc: 0.9897 - val_loss: 0.3629 - val_acc: 0.9159
Epoch 26/50
38400/38400 [==============================] - 50s 1ms/step - loss: 0.0
240 - acc: 0.9931 - val_loss: 0.3875 - val_acc: 0.9149
Epoch 27/50
38400/38400 [==============================] - 49s 1ms/step - loss: 0.0
321 - acc: 0.9898 - val_loss: 0.3851 - val_acc: 0.9121
Epoch 28/50
38400/38400 [==============================] - 49s 1ms/step - loss: 0.0
308 - acc: 0.9904 - val_loss: 0.3739 - val_acc: 0.9182
Epoch 29/50
38400/38400 [==============================] - 50s 1ms/step - loss: 0.0
206 - acc: 0.9943 - val_loss: 0.3936 - val_acc: 0.9200
Epoch 30/50
38400/38400 [==============================] - 50s 1ms/step - loss: 0.0
195 - acc: 0.9943 - val_loss: 0.4437 - val_acc: 0.9093
Epoch 31/50
38400/38400 [==============================] - 49s 1ms/step - loss: 0.0
208 - acc: 0.9936 - val_loss: 0.4031 - val_acc: 0.9160
Epoch 32/50
38400/38400 [==============================] - 49s 1ms/step - loss: 0.0
185 - acc: 0.9944 - val_loss: 0.4308 - val_acc: 0.9136
Epoch 33/50
38400/38400 [==============================] - 49s 1ms/step - loss: 0.0
125 - acc: 0.9971 - val_loss: 0.4180 - val_acc: 0.9182
Epoch 34/50
38400/38400 [==============================] - 50s 1ms/step - loss: 0.0
098 - acc: 0.9980 - val_loss: 0.4394 - val_acc: 0.9165
Epoch 35/50
38400/38400 [==============================] - 50s 1ms/step - loss: 0.0
096 - acc: 0.9977 - val_loss: 0.4343 - val_acc: 0.9191
Epoch 36/50
38400/38400 [==============================] - 49s 1ms/step - loss: 0.0
091 - acc: 0.9977 - val_loss: 0.4436 - val_acc: 0.9177
Epoch 37/50
```

```
38400/38400 [==============================] - 50s 1ms/step - loss: 0.0
256 - acc: 0.9916 - val_loss: 0.4829 - val_acc: 0.9082
Epoch 38/50
38400/38400 [==============================] - 50s 1ms/step - loss: 0.0
266 - acc: 0.9905 - val_loss: 0.4644 - val_acc: 0.9118
Epoch 39/50
38400/38400 [==============================] - 49s 1ms/step - loss: 0.0
161 - acc: 0.9949 - val_loss: 0.4784 - val_acc: 0.9165
Epoch 40/50
38400/38400 [==============================] - 49s 1ms/step - loss: 0.0
084 - acc: 0.9979 - val_loss: 0.4699 - val_acc: 0.9143
Epoch 41/50
38400/38400 [==============================] - 49s 1ms/step - loss: 0.0
084 - acc: 0.9976 - val_loss: 0.4645 - val_acc: 0.9150
Epoch 42/50
38400/38400 [==============================] - 49s 1ms/step - loss: 0.0
077 - acc: 0.9980 - val_loss: 0.4890 - val_acc: 0.9183
Epoch 43/50
38400/38400 [==============================] - 49s 1ms/step - loss: 0.0
037 - acc: 0.9994 - val_loss: 0.4833 - val_acc: 0.9156
Epoch 44/50
38400/38400 [==============================] - 49s 1ms/step - loss: 0.0
052 - acc: 0.9988 - val_loss: 0.4893 - val_acc: 0.9155
Epoch 45/50
38400/38400 [==============================] - 48s 1ms/step - loss: 0.0
311 - acc: 0.9889 - val_loss: 0.5105 - val_acc: 0.9089
Epoch 46/50
38400/38400 [==============================] - 49s 1ms/step - loss: 0.0
168 - acc: 0.9943 - val_loss: 0.4920 - val_acc: 0.9163
Epoch 47/50
38400/38400 [==============================] - 51s 1ms/step - loss: 0.0
045 - acc: 0.9992 - val_loss: 0.5021 - val_acc: 0.9173
Epoch 48/50
38400/38400 [==============================] - 50s 1ms/step - loss: 0.0
110 - acc: 0.9972 - val_loss: 0.5144 - val_acc: 0.9150
Epoch 49/50
38400/38400 [==============================] - 51s 1ms/step - loss: 0.0
181 - acc: 0.9952 - val_loss: 0.5338 - val_acc: 0.9117
Epoch 50/50
38400/38400 [==============================] - 49s 1ms/step - loss: 0.0
201 - acc: 0.9932 - val_loss: 0.4929 - val_acc: 0.9184
```




```
plt.plot(cnn_result.history['acc'])
plt.plot(cnn_result.history['val_acc'])
plt.title('Model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'validation'], loc='upper left')
plt.show()
# summarize history for loss
plt.plot(cnn_result.history['loss'])
plt.plot(cnn_result.history['val_loss'])
plt.title('Model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'validation'], loc='upper left')
plt.show()
```






```
test_result = cnn_model.evaluate(test_x,target_test_y, verbose=0)
print('Classification loss:', test_result[0])
print('Test Accuracy:', test_result[1])
```

```
Classification loss: 0.32
Test Accuracy: 0.915666
```

```
images,labels = ld.load('test')
test_x = images
test_x = test_x.reshape(test_x.shape[0], img_rows, img_cols, 1)
target_test_y = labels
test_x = test_x.astype('float32')
test_x = test_x / 255
test_y = to_categorical(target_test_y)

y_pred = cnn_model.predict(test_x)
y_pred = np.argmax(np.round(y_pred),axis=1)
from sklearn.metrics import classification_report
print(classification_report(target_test_y, y_pred))
cm = confusion_matrix(target_test_y, y_pred)
print(cm)
```

```
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]] [9 2 1 ... 8 1 5]
```

```
             precision    recall  f1-score   support

          0       0.85      0.83      0.84      1000
          1       1.00      0.96      0.98      1000
          2       0.84      0.86      0.85      1000
          3       0.87      0.92      0.90      1000
          4       0.86      0.85      0.86      1000
          5       0.98      0.96      0.97      1000
          6       0.76      0.74      0.75      1000
          7       0.95      0.98      0.96      1000
          8       0.97      0.97      0.97      1000
          9       0.97      0.97      0.97      1000

avg / total       0.91      0.90      0.90     10000

[[830   0  18  25   3   1 113   0   9   1]
 [  7 958   1  27   4   0   1   0   2   0]
 [ 13   0 864  10  47   0  64   0   2   0]
 [ 25   0  12 919  22   0  18   0   4   0]
 [  4   1  65  33 853   0  42   1   1   0]
 [  0   0   0   0   0 965   0  22   2  11]
 [ 90   0  67  31  62   0 744   0   6   0]
 [  0   0   0   0   0   7   0 977   1  15]
 [  8   0   2   8   3   1   2   5 970   1]
 [  0   0   0   0   0   6   1  24   1 968]]
```

```
y_classes = ["T-shirt/top","Trouser","Pullover","Dress","Coat","Sandal","Shirt","Sneaker","Bag","Ankle boot"]
import seaborn as sns
sns.heatmap(cm, annot=True, fmt='d',xticklabels=y_classes, yticklabels=y_classes)
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.show()
```