Human Activity Recongnition Using LSTM

October 23, 2018

1 Human Activity Recognition

2 Description

These days Smartphones have become an integral part of our life. We cannot assume our life without a mobile phone. Since, the advent of Smartphones, a revolution has been created in the mobile communication industry. Smartphones are not just restricted for calling these days. Infact, they are more often used for entertainment purpose.

Smartphone manufacturing companies load Smartphones with various sensors to enhance the user experinece. Two of the such sensors are Accelerometer and Gyroscope. Accelerometer measures acceleration while Gyroscope measures angular velocity.

Here, we will try to use the data provided by accelerometer and gyroscope of Smartphone to classify the activity which a Smartphone user is performing

3 Why this is Useful?

These days, in addition to Smartphones, we are also using Smart-Watches like Fitbit or Apple-Watch, which help us to track our health. They monitor our each activity throughout the day check how many calories we have burnt. How many hours have we slept. However, in addition to Accelerometer and Gyroscope, they also use Heart-Rate data to monitor our activity. Since, we only have Smartphone data so just by using Accelerometer and Gyroscope data we will monitor the activity of a person. This software can then be converted into an App which can be downloaded in Smartphone. Hence, a person who has Smartphone can monitor his/her health using this App

4 Information about Data

5 How Data is recorded

The experiments have been carried out with a group of 30 volunteers within an age bracket of 19-48 years. Each person performed six activities (WALKING, WALKING-UPSTAIRS, WALKING-DOWNSTAIRS, SITTING-DOWN, STANDING-UP, LAYING-DOWN) wearing a smartphone (Samsung Galaxy S II) on the waist. Using its embedded accelerometer and gyroscope, we captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz. The experiments have been video-recorded to label the data manually. The obtained dataset has been randomly partitioned into two sets, where 70% of the volunteers was selected for generating the training data and 30% the test data. 5.2. Features

These sensor signals are pre-processed by applying noise filters and then sampled in fixed-width windows(sliding windows) of 2.56 seconds each with 50% overlap. i.e., each window has 128 readings. A 128 size vector is created from each window. From Each window or to be more precise, from each 128 readings domain experts from signal processing have engineered feature vector of size 561 by calculating variables from the time and frequency domain. In our dataset, each data-point represents a window with different readings. 561 features are stored in the file "Features.docx". Check it out. Check out 561 features here.(In your blog give here the link of the docx file of features which you upload on github). The acceleration signal was separated into Body and Gravity acceleration signals(tBodyAcc-XYZ and tGravityAcc-XYZ) using some low pass filter with corner frequency of 0.3Hz. After that, the body linear acceleration and angular velocity were derived in time to obtain jerk signals (tBodyAccJerk-XYZ and tBodyGyroJerk-XYZ). The magnitude of these 3-dimensional signals were calculated using the Euclidian norm. This magnitudes are represented as features with names like tBodyAccMag, tGravityAccMag, tBodyAccJerkMag, tBodyGyroMag and tBodyGyroJerkMag. Finally, We've got frequency domain signals from some of the available signals by applying a FFT (Fast Fourier Transform). These signals obtained were labelled with prefix 'f' just like original signals with prefix 't'. These signals are labelled as fBodyAcc-XYZ, fBodyGyroMag etc.

These are the signals that we got so far.

tBodyAcc-XYZ tGravityAcc-XYZ tBodyAccJerk-XYZ tBodyGyro-XYZ tBodyGyroJerk-XYZ tBodyAccMag tGravityAccMag tBodyAccJerkMag tBodyGyroMag tBodyGyroJerkMag fBodyAcc-XYZ fBodyAccJerk-XYZ fBodyGyro-XYZ fBodyAccMag fBodyAccJerkMag fBodyGyroMag fBodyGyroJerkMag

9 We can estimate some set of variables from the above signals. i.e., We will estimate the following properties on each and every signal that we recorded so far.

mean(): Mean value std(): Standard deviation mad(): Median absolute deviation max(): Largest value in array min(): Smallest value in array sma(): Signal magnitude area energy(): Energy measure. Sum of the squares divided by the number of values. iqr(): Inter-quartile range entropy(): Signal entropy arCoeff(): Auto-regression coefficients with Burg order equal to 4 correlation(): correlation coefficient between two signals maxInds(): index of the frequency component with largest magnitude meanFreq(): Weighted average of the frequency components to obtain a mean frequency skewness(): skewness of the frequency domain signal kurtosis(): kurtosis of the frequency domain signal bandsEnergy(): Energy of a frequency interval within the 64 bins of the FFT of each window. angle(): Angle between to vectors.

10 We can obtain some other vectors by taking the average of signals in a single window sample. These are used on the angle() variable.

In [19]:
```
gravityMean
tBodyAccMean
tBodyAccJerkMean
tBodyGyroMean
tBodyGyroJerkMean
```

```
---------------------------------------------------------------------------

NameError                                 Traceback (most recent call last)
```

```
<ipython-input-19-d3f38673bd33> in <module>()
----> 1 gravityMean
      2 tBodyAccMean
      3 tBodyAccJerkMean
      4 tBodyGyroMean
      5 tBodyGyroJerkMean


NameError: name 'gravityMean' is not defined
```

6 Data Source

Data is downloaded from following source: https://archive.ics.uci.edu/ml/datasets/human+activity+recognitio

7 Quick Overview of Dataset

Accelerometer and Gyroscope readings are taken from 30 volunteers(referred as subjects) while performing the following 6 Activities.

These activites are encoded as follows: WALKING-- 1 WALKING_UPSTAIRS-- 2 WALKING_DOWNSTAIRS-- 3 SITTING-- 4 STANDING-- 5 LYING-- 6

1.Readings are divided into a window of 2.56 seconds with 50% overlapping. 2.Accelerometer readings are divided into gravity acceleration and body acceleration readings, which has x, y and z components each. 3.Gyroscope readings are the measure of angular velocities which has x, y and z components. 4.Jerk signals are calculated for Body-Acceleration readings. 5.Fourier Transforms are made on the above time readings to obtain frequency readings. 6.Now, on all the base signal readings., mean, max, mad, sma, arcoefficient, energy-bands, entropy etc., are calculated for each window. 7.Extra features are calculated by taking the average of signals in a single window sample. These are used on the angle() variable. 8.Finally, we got feature vector of 561 features and these features are given in the dataset. Each window of readings is a data-point of 561 features.

8 Y-Encoded Labels

In [ ]:
```
WALKING-- 1
WALKING_UPSTAIRS-- 2
WALKING_DOWNSTAIRS-- 3
SITTING-- 4
STANDING-- 5
LYING-- 6
```

9 Business Problem

Work-flow is as follows:

1.Domain experts from the field of Signal Processing collects the data from Accelerometer and Gyroscope of Smartphone. 2.They break up the data in the time window of 2.56 seconds with 50% overlapping i.e., 128 reading 3.They engineered 561 features from each time window of 2.56 seconds.

By using either human engineered 561 feature data or raw features of 128 reading, our goal is to predict one of the six activities that a Smartphone user is performing at that 2.56 Seconds time window.

10 Problem Statement

By using either human engineered 561 feature data or raw features of 128 reading, our goal is to predict one of the six activities that a Smartphone user is performing at that 2.56 Seconds time window.

11 Objective and Constraints

1.No Low latency requirement. 2.Errors are not much costly.

All of the Accelerometer and Gyroscope are tri-axial, means that they measure acceleration and angular-velocity respectively in all the three axis namely X-axis, Y-axis and Z-axis. So, we have in total six time-series data. Given this six time-series data, we want to predict six activities namely Walking or Walking-Upstairs or Walking-Downstairs or Lying-Down or Standing-Up or Sitting-Down.

At the outset, this is a multi-class classification problem.

12 ML Problem Formulation

All of the Accelerometer and Gyroscope are tri-axial, means that they measure acceleration and angular-velocity respectively in all the three axis namely X-axis, Y-axis and Z-axis. So, we have in total six time-series data. Given this six time-series data, we want to predict six activities namely Walking or Walking-Upstairs or Walking-Downstairs or Lying-Down or Standing-Up or Sitting-Down.

At the outset, this is a multi-class classification problem.

13 Performance Metric

1.We will use Accuracy as one of the metric. 2.We will also use Confusion-Matrix to check that in which two activities our model is confused and predicting incorrect activity. For example, between Standing-Up and Sitting-Down. Between Walking-Upstairs and Walking-Downstairs

14 Data

All the data is present in 'UCI_HAR_dataset/' folder in present working directory. Feature names are present in 'UCI_HAR_dataset/features.txt'

Train Data 'UCI_HAR_dataset/train/X_train.txt' 'UCI_HAR_dataset/train/subject_train.txt' 'UCI_HAR_dataset/train/y_train.txt' Test Data 'UCI_HAR_dataset/test/X_test.txt' 'UCI_HAR_dataset/test/subject_test.txt' 'UCI_HAR_dataset/test/y_test.txt'

15 Data-Points Distribution

1.30 test-subjects data is randomly split to 70%(21) train and 30%(7) test data. 2.Each data-point corresponds one of the 6 Activities.

16 Plan of Action

We will apply classical Machine Learning models on these 561 sized domain expert engineered features. As we know that LSTM works well on time-series data, so we have decided that we will apply LSTM of Recurrent Neural Networks on 128 sized raw readings that we obtained from accelerometer and gyroscope signals.

```
In [ ]: # Importing Libraries
```

```
In [1]: import pandas as pd
        import numpy as np
```

```
In [2]: # Activities are the class labels
        # It is a 6 class classification
        ACTIVITIES = {
            0: 'WALKING',
            1: 'WALKING_UPSTAIRS',
            2: 'WALKING_DOWNSTAIRS',
            3: 'SITTING',
            4: 'STANDING',
            5: 'LAYING',
        }

        # Utility function to print the confusion matrix
        def confusion_matrix(Y_true, Y_pred):
            Y_true = pd.Series([ACTIVITIES[y] for y in np.argmax(Y_true, axis=1)])
            Y_pred = pd.Series([ACTIVITIES[y] for y in np.argmax(Y_pred, axis=1)])

            return pd.crosstab(Y_true, Y_pred, rownames=['True'], colnames=['Pred'])
```

16.0.1 Data

```
In [3]: # Data directory
        DATADIR = 'UCI_HAR_Dataset'
```

```
In [4]: # Raw data signals
        # Signals are from Accelerometer and Gyroscope
        # The signals are in x,y,z directions
        # Sensor signals are filtered to have only body acceleration
```

```
# excluding the acceleration due to gravity
# Triaxial acceleration from the accelerometer is total acceleration
SIGNALS = [
    "body_acc_x",
    "body_acc_y",
    "body_acc_z",
    "body_gyro_x",
    "body_gyro_y",
    "body_gyro_z",
    "total_acc_x",
    "total_acc_y",
    "total_acc_z"
]
```

In [5]:
```
# Utility function to read the data from csv file
def _read_csv(filename):
    return pd.read_csv(filename, delim_whitespace=True, header=None)

# Utility function to load the load
def load_signals(subset):
    signals_data = []

    for signal in SIGNALS:
        filename = f'UCI_HAR_Dataset/{subset}/Inertial Signals/{signal}_{subset}.txt'
        signals_data.append(
            _read_csv(filename).as_matrix()
        )

    # Transpose is used to change the dimensionality of the output,
    # aggregating the signals by combination of sample/timestep.
    # Resultant shape is (7352 train/2947 test samples, 128 timesteps, 9 signals)
    return np.transpose(signals_data, (1, 2, 0))
```

In [6]:
```
def load_y(subset):
    """
    The objective that we are trying to predict is a integer, from 1 to 6,
    that represents a human activity. We return a binary representation of
    every sample objective as a 6 bits vector using One Hot Encoding
    (https://pandas.pydata.org/pandas-docs/stable/generated/pandas.get_dummies.html)
    """
    filename = f'UCI_HAR_Dataset/{subset}/y_{subset}.txt'
    y = _read_csv(filename)[0]

    return pd.get_dummies(y).as_matrix()
```

In [7]:
```
def load_data():
    """
    Obtain the dataset from multiple files.
```

```
        Returns: X_train, X_test, y_train, y_test
        """
        X_train, X_test = load_signals('train'), load_signals('test')
        y_train, y_test = load_y('train'), load_y('test')

        return X_train, X_test, y_train, y_test
```

In [8]:
```
# Importing tensorflow
np.random.seed(42)
import tensorflow as tf
tf.set_random_seed(42)
```

```
C:\Users\Saurabh\Anaconda3\lib\site-packages\h5py\__init__.py:36: FutureWarning: Conversion of
  from ._conv import register_converters as _register_converters
```

In [9]:
```
# Configuring a session
session_conf = tf.ConfigProto(
    intra_op_parallelism_threads=1,
    inter_op_parallelism_threads=1
)
```

In [10]:
```
# Import Keras
from keras import backend as K
sess = tf.Session(graph=tf.get_default_graph(), config=session_conf)
K.set_session(sess)
```

```
Using TensorFlow backend.
```

In [11]:
```
# Importing libraries
from keras.models import Sequential
from keras.layers import LSTM
from keras.layers.core import Dense, Dropout
```

In [30]:
```
# Initializing parameters
epochs = 300
batch_size = 32
n_hidden = 64
```

In [31]:
```
# Utility function to count the number of classes
def _count_classes(y):
    return len(set([tuple(category) for category in y]))
```

In [32]:
```
# Loading the train and test data
X_train, X_test, Y_train, Y_test = load_data()
```

```
C:\Users\Saurabh\Anaconda3\lib\site-packages\ipykernel_launcher.py:12: FutureWarning: Method .a
  if sys.path[0] == '':
```

```
In [33]: timesteps = len(X_train[0])
         input_dim = len(X_train[0][0])
         n_classes = _count_classes(Y_train)

         print(timesteps)
         print(input_dim)
         print(len(X_train))
```

```
128
9
7352
```

- Defining the Architecture of LSTM

```
In [34]: # Initiliazing the sequential model
         model = Sequential()
         # Configuring the parameters
         model.add(LSTM(n_hidden, input_shape=(timesteps, input_dim)))
         # Adding a dropout layer
         model.add(Dropout(0.5))
         # Adding a dense output layer with sigmoid activation
         model.add(Dense(n_classes, activation='sigmoid'))
         model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_3 (LSTM)                (None, 64)                18944
_________________________________________________________________
dropout_3 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_3 (Dense)              (None, 6)                 390
=================================================================
Total params: 19,334
Trainable params: 19,334
Non-trainable params: 0
_________________________________________________________________
```

```
In [35]: # Compiling the model
         model.compile(loss='categorical_crossentropy',
                       optimizer='adam',
                       metrics=['accuracy'])
```

```
In [36]: # Training the model
         model.fit(X_train,
                   Y_train,
                   batch_size=batch_size,
```

```
                   validation_data=(X_test, Y_test),
                   epochs=epochs)

Train on 7352 samples, validate on 2947 samples
Epoch 1/300
7352/7352 [==============================] - 24s 3ms/step - loss: 1.4284 - acc: 0.3644 - val_lo
Epoch 2/300
7352/7352 [==============================] - 22s 3ms/step - loss: 1.3148 - acc: 0.3923 - val_lo
Epoch 3/300
7352/7352 [==============================] - 24s 3ms/step - loss: 1.2377 - acc: 0.4463 - val_lo
Epoch 4/300
7352/7352 [==============================] - 23s 3ms/step - loss: 1.1586 - acc: 0.4737 - val_lo
Epoch 5/300
7352/7352 [==============================] - 22s 3ms/step - loss: 1.2517 - acc: 0.4286 - val_lo
Epoch 6/300
7352/7352 [==============================] - 22s 3ms/step - loss: 1.1608 - acc: 0.5076 - val_lo
Epoch 7/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.9851 - acc: 0.5613 - val_lo
Epoch 8/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.9568 - acc: 0.5768 - val_lo
Epoch 9/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.8428 - acc: 0.6217 - val_lo
Epoch 10/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.7760 - acc: 0.6328 - val_lo
Epoch 11/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.7391 - acc: 0.6506 - val_lo
Epoch 12/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.7065 - acc: 0.6737 - val_lo
Epoch 13/300
7352/7352 [==============================] - 23s 3ms/step - loss: 0.5739 - acc: 0.7421 - val_lo
Epoch 14/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.5620 - acc: 0.7606 - val_lo
Epoch 15/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.4878 - acc: 0.7818 - val_lo
Epoch 16/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.6082 - acc: 0.7448 - val_lo
Epoch 17/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.8085 - acc: 0.6663 - val_lo
Epoch 18/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.5122 - acc: 0.7990 - val_lo
Epoch 19/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.6121 - acc: 0.7682 - val_lo
Epoch 20/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.5246 - acc: 0.8075 - val_lo
Epoch 21/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.3587 - acc: 0.8811 - val_lo
Epoch 22/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.4166 - acc: 0.8690 - val_lo
```

```
Epoch 23/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.3675 - acc: 0.8950 - val_lo
Epoch 24/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.5638 - acc: 0.7954 - val_lo
Epoch 25/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.3414 - acc: 0.8794 - val_lo
Epoch 26/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.2948 - acc: 0.9036 - val_lo
Epoch 27/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.2557 - acc: 0.9112 - val_lo
Epoch 28/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.2705 - acc: 0.9108 - val_lo
Epoch 29/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.2208 - acc: 0.9187 - val_lo
Epoch 30/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.2366 - acc: 0.9188 - val_lo
Epoch 31/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1712 - acc: 0.9377 - val_lo
Epoch 32/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1859 - acc: 0.9323 - val_lo
Epoch 33/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1516 - acc: 0.9415 - val_lo
Epoch 34/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.4006 - acc: 0.8566 - val_lo
Epoch 35/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.2925 - acc: 0.8991 - val_lo
Epoch 36/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1952 - acc: 0.9338 - val_lo
Epoch 37/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1562 - acc: 0.9423 - val_lo
Epoch 38/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.2587 - acc: 0.9127 - val_lo
Epoch 39/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1869 - acc: 0.9368 - val_lo
Epoch 40/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1435 - acc: 0.9457 - val_lo
Epoch 41/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1460 - acc: 0.9445 - val_lo
Epoch 42/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1453 - acc: 0.9460 - val_lo
Epoch 43/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1309 - acc: 0.9476 - val_lo
Epoch 44/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1829 - acc: 0.9339 - val_lo
Epoch 45/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1778 - acc: 0.9381 - val_lo
Epoch 46/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.2386 - acc: 0.9240 - val_lo
```

```
Epoch 47/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.2023 - acc: 0.9279 - val_lo
Epoch 48/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1627 - acc: 0.9467 - val_lo
Epoch 49/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1717 - acc: 0.9430 - val_lo
Epoch 50/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1299 - acc: 0.9520 - val_lo
Epoch 51/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1379 - acc: 0.9468 - val_lo
Epoch 52/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1245 - acc: 0.9482 - val_lo
Epoch 53/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1499 - acc: 0.9429 - val_lo
Epoch 54/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1514 - acc: 0.9408 - val_lo
Epoch 55/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.2301 - acc: 0.9230 - val_lo
Epoch 56/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1780 - acc: 0.9314 - val_lo
Epoch 57/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1400 - acc: 0.9484 - val_lo
Epoch 58/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.2840 - acc: 0.9112 - val_lo
Epoch 59/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1785 - acc: 0.9389 - val_lo
Epoch 60/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1423 - acc: 0.9491 - val_lo
Epoch 61/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1715 - acc: 0.9368 - val_lo
Epoch 62/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1732 - acc: 0.9410 - val_lo
Epoch 63/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1449 - acc: 0.9468 - val_lo
Epoch 64/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1437 - acc: 0.9476 - val_lo
Epoch 65/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1262 - acc: 0.9518 - val_lo
Epoch 66/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1207 - acc: 0.9532 - val_lo
Epoch 67/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1532 - acc: 0.9440 - val_lo
Epoch 68/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1256 - acc: 0.9493 - val_lo
Epoch 69/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1167 - acc: 0.9540 - val_lo
Epoch 70/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1373 - acc: 0.9471 - val_lo
```

```
Epoch 71/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1417 - acc: 0.9406 - val_lo
Epoch 72/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1263 - acc: 0.9441 - val_lo
Epoch 73/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1319 - acc: 0.9472 - val_lo
Epoch 74/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1806 - acc: 0.9400 - val_lo
Epoch 75/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1956 - acc: 0.9385 - val_lo
Epoch 76/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1451 - acc: 0.9455 - val_lo
Epoch 77/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1225 - acc: 0.9480 - val_lo
Epoch 78/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1197 - acc: 0.9501 - val_lo
Epoch 79/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1202 - acc: 0.9514 - val_lo
Epoch 80/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1661 - acc: 0.9387 - val_lo
Epoch 81/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1982 - acc: 0.9357 - val_lo
Epoch 82/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1338 - acc: 0.9509 - val_lo
Epoch 83/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1176 - acc: 0.9551 - val_lo
Epoch 84/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.2776 - acc: 0.9161 - val_lo
Epoch 85/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1763 - acc: 0.9359 - val_lo
Epoch 86/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1629 - acc: 0.9347 - val_lo
Epoch 87/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1494 - acc: 0.9400 - val_lo
Epoch 88/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1376 - acc: 0.9463 - val_lo
Epoch 89/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1353 - acc: 0.9456 - val_lo
Epoch 90/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1405 - acc: 0.9465 - val_lo
Epoch 91/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1277 - acc: 0.9490 - val_lo
Epoch 92/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1315 - acc: 0.9467 - val_lo
Epoch 93/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1290 - acc: 0.9470 - val_lo
Epoch 94/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1273 - acc: 0.9502 - val_lo
```

```
Epoch 95/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1315 - acc: 0.9468 - val_lo
Epoch 96/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.2131 - acc: 0.9305 - val_lo
Epoch 97/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1738 - acc: 0.9395 - val_lo
Epoch 98/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1344 - acc: 0.9512 - val_lo
Epoch 99/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1254 - acc: 0.9525 - val_lo
Epoch 100/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1246 - acc: 0.9517 - val_lo
Epoch 101/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1217 - acc: 0.9528 - val_lo
Epoch 102/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1160 - acc: 0.9535 - val_lo
Epoch 103/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1241 - acc: 0.9493 - val_lo
Epoch 104/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1266 - acc: 0.9464 - val_lo
Epoch 105/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1156 - acc: 0.9548 - val_lo
Epoch 106/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1168 - acc: 0.9533 - val_lo
Epoch 107/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1134 - acc: 0.9563 - val_lo
Epoch 108/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1260 - acc: 0.9538 - val_lo
Epoch 109/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1281 - acc: 0.9513 - val_lo
Epoch 110/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1240 - acc: 0.9523 - val_lo
Epoch 111/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1355 - acc: 0.9508 - val_lo
Epoch 112/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1168 - acc: 0.9544 - val_lo
Epoch 113/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1221 - acc: 0.9528 - val_lo
Epoch 114/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1194 - acc: 0.9551 - val_lo
Epoch 115/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1147 - acc: 0.9570 - val_lo
Epoch 116/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1139 - acc: 0.9554 - val_lo
Epoch 117/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1105 - acc: 0.9548 - val_lo
Epoch 118/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1476 - acc: 0.9427 - val_lo
```

```
Epoch 119/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1338 - acc: 0.9421 - val_lo
Epoch 120/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1197 - acc: 0.9505 - val_lo
Epoch 121/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1667 - acc: 0.9380 - val_lo
Epoch 122/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1307 - acc: 0.9463 - val_lo
Epoch 123/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1149 - acc: 0.9516 - val_lo
Epoch 124/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1127 - acc: 0.9512 - val_lo
Epoch 125/300
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1130 - acc: 0.9529 - val_lo
Epoch 126/300
7352/7352 [==============================] - 1253s 170ms/step - loss: 0.1120 - acc: 0.9557 - va
Epoch 127/300
7352/7352 [==============================] - 43s 6ms/step - loss: 0.1275 - acc: 0.9495 - val_lo
Epoch 128/300
7352/7352 [==============================] - 26s 4ms/step - loss: 0.1185 - acc: 0.9506 - val_lo
Epoch 129/300
7352/7352 [==============================] - 39s 5ms/step - loss: 0.1074 - acc: 0.9547 - val_lo
Epoch 130/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1272 - acc: 0.9484 - val_lo
Epoch 131/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1192 - acc: 0.9499 - val_lo
Epoch 132/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1190 - acc: 0.9538 - val_lo
Epoch 133/300
7352/7352 [==============================] - 38s 5ms/step - loss: 0.1170 - acc: 0.9504 - val_lo
Epoch 134/300
7352/7352 [==============================] - 42s 6ms/step - loss: 0.1171 - acc: 0.9529 - val_lo
Epoch 135/300
7352/7352 [==============================] - 43s 6ms/step - loss: 0.1133 - acc: 0.9533 - val_lo
Epoch 136/300
7352/7352 [==============================] - 42s 6ms/step - loss: 0.1079 - acc: 0.9524 - val_lo
Epoch 137/300
7352/7352 [==============================] - 43s 6ms/step - loss: 0.1284 - acc: 0.9490 - val_lo
Epoch 138/300
7352/7352 [==============================] - 43s 6ms/step - loss: 0.1229 - acc: 0.9531 - val_lo
Epoch 139/300
7352/7352 [==============================] - 43s 6ms/step - loss: 0.1146 - acc: 0.9544 - val_lo
Epoch 140/300
7352/7352 [==============================] - 41s 6ms/step - loss: 0.1213 - acc: 0.9514 - val_lo
Epoch 141/300
7352/7352 [==============================] - 42s 6ms/step - loss: 0.1169 - acc: 0.9557 - val_lo
Epoch 142/300
7352/7352 [==============================] - 35s 5ms/step - loss: 0.1123 - acc: 0.9581 - val_lo
```

```
Epoch 143/300
7352/7352 [==============================] - 38s 5ms/step - loss: 0.1184 - acc: 0.9513 - val_lo
Epoch 144/300
7352/7352 [==============================] - 34s 5ms/step - loss: 0.1242 - acc: 0.9461 - val_lo
Epoch 145/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1088 - acc: 0.9529 - val_lo
Epoch 146/300
7352/7352 [==============================] - 44s 6ms/step - loss: 0.1202 - acc: 0.9502 - val_lo
Epoch 147/300
7352/7352 [==============================] - 36s 5ms/step - loss: 0.1097 - acc: 0.9581 - val_lo
Epoch 148/300
7352/7352 [==============================] - 33s 4ms/step - loss: 0.1061 - acc: 0.9576 - val_lo
Epoch 149/300
7352/7352 [==============================] - 33s 4ms/step - loss: 0.1094 - acc: 0.9555 - val_lo
Epoch 150/300
7352/7352 [==============================] - 34s 5ms/step - loss: 0.1082 - acc: 0.9566 - val_lo
Epoch 151/300
7352/7352 [==============================] - 34s 5ms/step - loss: 0.1025 - acc: 0.9592 - val_lo
Epoch 152/300
7352/7352 [==============================] - 36s 5ms/step - loss: 0.1123 - acc: 0.9538 - val_lo
Epoch 153/300
7352/7352 [==============================] - 45s 6ms/step - loss: 0.1685 - acc: 0.9403 - val_lo
Epoch 154/300
7352/7352 [==============================] - 42s 6ms/step - loss: 0.1104 - acc: 0.9525 - val_lo
Epoch 155/300
7352/7352 [==============================] - 38s 5ms/step - loss: 0.1115 - acc: 0.9553 - val_lo
Epoch 156/300
7352/7352 [==============================] - 33s 5ms/step - loss: 0.1192 - acc: 0.9527 - val_lo
Epoch 157/300
7352/7352 [==============================] - 33s 4ms/step - loss: 0.1104 - acc: 0.9597 - val_lo
Epoch 158/300
7352/7352 [==============================] - 40s 6ms/step - loss: 0.1098 - acc: 0.9574 - val_lo
Epoch 159/300
7352/7352 [==============================] - 35s 5ms/step - loss: 0.1059 - acc: 0.9597 - val_lo
Epoch 160/300
7352/7352 [==============================] - 34s 5ms/step - loss: 0.1320 - acc: 0.9453 - val_lo
Epoch 161/300
7352/7352 [==============================] - 34s 5ms/step - loss: 0.1115 - acc: 0.9499 - val_lo
Epoch 162/300
7352/7352 [==============================] - 35s 5ms/step - loss: 0.1088 - acc: 0.9548 - val_lo
Epoch 163/300
7352/7352 [==============================] - 33s 5ms/step - loss: 0.1092 - acc: 0.9536 - val_lo
Epoch 164/300
7352/7352 [==============================] - 42s 6ms/step - loss: 0.1108 - acc: 0.9533 - val_lo
Epoch 165/300
7352/7352 [==============================] - 39s 5ms/step - loss: 0.1346 - acc: 0.9416 - val_lo
Epoch 166/300
7352/7352 [==============================] - 39s 5ms/step - loss: 0.1300 - acc: 0.9486 - val_lo
```

```
Epoch 167/300
7352/7352 [==============================] - 41s 6ms/step - loss: 0.1103 - acc: 0.9582 - val_lo
Epoch 168/300
7352/7352 [==============================] - 38s 5ms/step - loss: 0.1086 - acc: 0.9563 - val_lo
Epoch 169/300
7352/7352 [==============================] - 39s 5ms/step - loss: 0.1089 - acc: 0.9576 - val_lo
Epoch 170/300
7352/7352 [==============================] - 39s 5ms/step - loss: 0.1183 - acc: 0.9574 - val_lo
Epoch 171/300
7352/7352 [==============================] - 39s 5ms/step - loss: 0.1005 - acc: 0.9574 - val_lo
Epoch 172/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1005 - acc: 0.9578 - val_lo
Epoch 173/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1092 - acc: 0.9588 - val_lo
Epoch 174/300
7352/7352 [==============================] - 38s 5ms/step - loss: 0.1034 - acc: 0.9597 - val_lo
Epoch 175/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1028 - acc: 0.9561 - val_lo
Epoch 176/300
7352/7352 [==============================] - 38s 5ms/step - loss: 0.1072 - acc: 0.9561 - val_lo
Epoch 177/300
7352/7352 [==============================] - 38s 5ms/step - loss: 0.1093 - acc: 0.9557 - val_lo
Epoch 178/300
7352/7352 [==============================] - 38s 5ms/step - loss: 0.1101 - acc: 0.9555 - val_lo
Epoch 179/300
7352/7352 [==============================] - 38s 5ms/step - loss: 0.1004 - acc: 0.9573 - val_lo
Epoch 180/300
7352/7352 [==============================] - 38s 5ms/step - loss: 0.1120 - acc: 0.9547 - val_lo
Epoch 181/300
7352/7352 [==============================] - 34s 5ms/step - loss: 0.1104 - acc: 0.9510 - val_lo
Epoch 182/300
7352/7352 [==============================] - 33s 5ms/step - loss: 0.1223 - acc: 0.9531 - val_lo
Epoch 183/300
7352/7352 [==============================] - 33s 4ms/step - loss: 0.1390 - acc: 0.9354 - val_lo
Epoch 184/300
7352/7352 [==============================] - 33s 5ms/step - loss: 0.1504 - acc: 0.9373 - val_lo
Epoch 185/300
7352/7352 [==============================] - 33s 5ms/step - loss: 0.1139 - acc: 0.9502 - val_lo
Epoch 186/300
7352/7352 [==============================] - 32s 4ms/step - loss: 0.1104 - acc: 0.9514 - val_lo
Epoch 187/300
7352/7352 [==============================] - 33s 4ms/step - loss: 0.1100 - acc: 0.9524 - val_lo
Epoch 188/300
7352/7352 [==============================] - 33s 5ms/step - loss: 0.1043 - acc: 0.9528 - val_lo
Epoch 189/300
7352/7352 [==============================] - 37s 5ms/step - loss: 0.1056 - acc: 0.9540 - val_lo
Epoch 190/300
7352/7352 [==============================] - 41s 6ms/step - loss: 0.1038 - acc: 0.9535 - val_lo
```

```
Epoch 191/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1106 - acc: 0.9566 - val_lo
Epoch 192/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.0998 - acc: 0.9584 - val_lo
Epoch 193/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.0995 - acc: 0.9591 - val_lo
Epoch 194/300
7352/7352 [==============================] - 39s 5ms/step - loss: 0.0993 - acc: 0.9600 - val_lo
Epoch 195/300
7352/7352 [==============================] - 39s 5ms/step - loss: 0.0953 - acc: 0.9596 - val_lo
Epoch 196/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1004 - acc: 0.9588 - val_lo
Epoch 197/300
7352/7352 [==============================] - 39s 5ms/step - loss: 0.1170 - acc: 0.9572 - val_lo
Epoch 198/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1590 - acc: 0.9431 - val_lo
Epoch 199/300
7352/7352 [==============================] - 44s 6ms/step - loss: 0.1601 - acc: 0.9457 - val_lo
Epoch 200/300
7352/7352 [==============================] - 41s 6ms/step - loss: 0.1212 - acc: 0.9551 - val_lo
Epoch 201/300
7352/7352 [==============================] - 41s 6ms/step - loss: 0.1406 - acc: 0.9517 - val_lo
Epoch 202/300
7352/7352 [==============================] - 42s 6ms/step - loss: 0.1190 - acc: 0.9570 - val_lo
Epoch 203/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1104 - acc: 0.9582 - val_lo
Epoch 204/300
7352/7352 [==============================] - 34s 5ms/step - loss: 0.1087 - acc: 0.9580 - val_lo
Epoch 205/300
7352/7352 [==============================] - 34s 5ms/step - loss: 0.1143 - acc: 0.9529 - val_lo
Epoch 206/300
7352/7352 [==============================] - 37s 5ms/step - loss: 0.1066 - acc: 0.9578 - val_lo
Epoch 207/300
7352/7352 [==============================] - 39s 5ms/step - loss: 0.1061 - acc: 0.9572 - val_lo
Epoch 208/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1048 - acc: 0.9588 - val_lo
Epoch 209/300
7352/7352 [==============================] - 42s 6ms/step - loss: 0.1412 - acc: 0.9523 - val_lo
Epoch 210/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1328 - acc: 0.9517 - val_lo
Epoch 211/300
7352/7352 [==============================] - 36s 5ms/step - loss: 0.1099 - acc: 0.9567 - val_lo
Epoch 212/300
7352/7352 [==============================] - 42s 6ms/step - loss: 0.1221 - acc: 0.9512 - val_lo
Epoch 213/300
7352/7352 [==============================] - 35s 5ms/step - loss: 0.1106 - acc: 0.9563 - val_lo
Epoch 214/300
7352/7352 [==============================] - 41s 6ms/step - loss: 0.1044 - acc: 0.9558 - val_lo
```

```
Epoch 215/300
7352/7352 [==============================] - 41s 6ms/step - loss: 0.1018 - acc: 0.9577 - val_lo
Epoch 216/300
7352/7352 [==============================] - 41s 6ms/step - loss: 0.1019 - acc: 0.9566 - val_lo
Epoch 217/300
7352/7352 [==============================] - 41s 6ms/step - loss: 0.1106 - acc: 0.9529 - val_lo
Epoch 218/300
7352/7352 [==============================] - 41s 6ms/step - loss: 0.1015 - acc: 0.9589 - val_lo
Epoch 219/300
7352/7352 [==============================] - 41s 6ms/step - loss: 0.1036 - acc: 0.9589 - val_lo
Epoch 220/300
7352/7352 [==============================] - 37s 5ms/step - loss: 0.0998 - acc: 0.9595 - val_lo
Epoch 221/300
7352/7352 [==============================] - 41s 6ms/step - loss: 0.1021 - acc: 0.9577 - val_lo
Epoch 222/300
7352/7352 [==============================] - 38s 5ms/step - loss: 0.1232 - acc: 0.9478 - val_lo
Epoch 223/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1313 - acc: 0.9422 - val_lo
Epoch 224/300
7352/7352 [==============================] - 41s 6ms/step - loss: 0.1422 - acc: 0.9404 - val_lo
Epoch 225/300
7352/7352 [==============================] - 40s 6ms/step - loss: 0.1183 - acc: 0.9465 - val_lo
Epoch 226/300
7352/7352 [==============================] - 41s 6ms/step - loss: 0.1113 - acc: 0.9495 - val_lo
Epoch 227/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1080 - acc: 0.9542 - val_lo
Epoch 228/300
7352/7352 [==============================] - 39s 5ms/step - loss: 0.1227 - acc: 0.9482 - val_lo
Epoch 229/300
7352/7352 [==============================] - 39s 5ms/step - loss: 0.1321 - acc: 0.9415 - val_lo
Epoch 230/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1082 - acc: 0.9520 - val_lo
Epoch 231/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1069 - acc: 0.9565 - val_lo
Epoch 232/300
7352/7352 [==============================] - 41s 6ms/step - loss: 0.1031 - acc: 0.9573 - val_lo
Epoch 233/300
7352/7352 [==============================] - 39s 5ms/step - loss: 0.1065 - acc: 0.9540 - val_lo
Epoch 234/300
7352/7352 [==============================] - 40s 6ms/step - loss: 0.1051 - acc: 0.9558 - val_lo
Epoch 235/300
7352/7352 [==============================] - 41s 6ms/step - loss: 0.1076 - acc: 0.9543 - val_lo
Epoch 236/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1107 - acc: 0.9567 - val_lo
Epoch 237/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1041 - acc: 0.9589 - val_lo
Epoch 238/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1127 - acc: 0.9548 - val_lo
```

```
Epoch 239/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1108 - acc: 0.9577 - val_lo
Epoch 240/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1044 - acc: 0.9584 - val_lo
Epoch 241/300
7352/7352 [==============================] - 39s 5ms/step - loss: 0.1019 - acc: 0.9593 - val_lo
Epoch 242/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.0995 - acc: 0.9580 - val_lo
Epoch 243/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1423 - acc: 0.9482 - val_lo
Epoch 244/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1149 - acc: 0.9525 - val_lo
Epoch 245/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1112 - acc: 0.9521 - val_lo
Epoch 246/300
7352/7352 [==============================] - 41s 6ms/step - loss: 0.1070 - acc: 0.9562 - val_lo
Epoch 247/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1038 - acc: 0.9554 - val_lo
Epoch 248/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1011 - acc: 0.9569 - val_lo
Epoch 249/300
7352/7352 [==============================] - 41s 6ms/step - loss: 0.0997 - acc: 0.9592 - val_lo
Epoch 250/300
7352/7352 [==============================] - 41s 6ms/step - loss: 0.0953 - acc: 0.9595 - val_lo
Epoch 251/300
7352/7352 [==============================] - 41s 6ms/step - loss: 0.0965 - acc: 0.9599 - val_lo
Epoch 252/300
7352/7352 [==============================] - 39s 5ms/step - loss: 0.1134 - acc: 0.9539 - val_lo
Epoch 253/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1013 - acc: 0.9581 - val_lo
Epoch 254/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.0947 - acc: 0.9606 - val_lo
Epoch 255/300
7352/7352 [==============================] - 39s 5ms/step - loss: 0.0995 - acc: 0.9589 - val_lo
Epoch 256/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.0969 - acc: 0.9591 - val_lo
Epoch 257/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.0946 - acc: 0.9585 - val_lo
Epoch 258/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1006 - acc: 0.9569 - val_lo
Epoch 259/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1617 - acc: 0.9434 - val_lo
Epoch 260/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1136 - acc: 0.9527 - val_lo
Epoch 261/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1338 - acc: 0.9509 - val_lo
Epoch 262/300
7352/7352 [==============================] - 39s 5ms/step - loss: 0.1070 - acc: 0.9569 - val_lo
```

```
Epoch 263/300
7352/7352 [==============================] - 39s 5ms/step - loss: 0.1031 - acc: 0.9559 - val_lo
Epoch 264/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.0982 - acc: 0.9573 - val_lo
Epoch 265/300
7352/7352 [==============================] - 39s 5ms/step - loss: 0.0991 - acc: 0.9578 - val_lo
Epoch 266/300
7352/7352 [==============================] - 40s 5ms/step - loss: 0.1367 - acc: 0.9411 - val_lo
Epoch 267/300
7352/7352 [==============================] - 40s 6ms/step - loss: 0.1390 - acc: 0.9418 - val_lo
Epoch 268/300
7352/7352 [==============================] - 34s 5ms/step - loss: 0.1270 - acc: 0.9441 - val_lo
Epoch 269/300
7352/7352 [==============================] - 32s 4ms/step - loss: 0.1168 - acc: 0.9453 - val_lo
Epoch 270/300
7352/7352 [==============================] - 33s 4ms/step - loss: 0.1128 - acc: 0.9487 - val_lo
Epoch 271/300
7352/7352 [==============================] - 32s 4ms/step - loss: 0.1107 - acc: 0.9527 - val_lo
Epoch 272/300
7352/7352 [==============================] - 32s 4ms/step - loss: 0.1079 - acc: 0.9512 - val_lo
Epoch 273/300
7352/7352 [==============================] - 32s 4ms/step - loss: 0.1077 - acc: 0.9517 - val_lo
Epoch 274/300
7352/7352 [==============================] - 34s 5ms/step - loss: 0.1112 - acc: 0.9548 - val_lo
Epoch 275/300
7352/7352 [==============================] - 39s 5ms/step - loss: 0.1090 - acc: 0.9529 - val_lo
Epoch 276/300
7352/7352 [==============================] - 39s 5ms/step - loss: 0.1058 - acc: 0.9544 - val_lo
Epoch 277/300
7352/7352 [==============================] - 34s 5ms/step - loss: 0.1123 - acc: 0.9528 - val_lo
Epoch 278/300
7352/7352 [==============================] - 33s 5ms/step - loss: 0.1090 - acc: 0.9550 - val_lo
Epoch 279/300
7352/7352 [==============================] - 36s 5ms/step - loss: 0.1832 - acc: 0.9363 - val_lo
Epoch 280/300
7352/7352 [==============================] - 37s 5ms/step - loss: 0.1165 - acc: 0.9506 - val_lo
Epoch 281/300
7352/7352 [==============================] - 35s 5ms/step - loss: 0.1097 - acc: 0.9525 - val_lo
Epoch 282/300
7352/7352 [==============================] - 34s 5ms/step - loss: 0.1055 - acc: 0.9539 - val_lo
Epoch 283/300
7352/7352 [==============================] - 36s 5ms/step - loss: 0.1048 - acc: 0.9558 - val_lo
Epoch 284/300
7352/7352 [==============================] - 36s 5ms/step - loss: 0.1137 - acc: 0.9538 - val_lo
Epoch 285/300
7352/7352 [==============================] - 36s 5ms/step - loss: 0.1051 - acc: 0.9569 - val_lo
Epoch 286/300
7352/7352 [==============================] - 35s 5ms/step - loss: 0.1029 - acc: 0.9585 - val_lo
```

```
Epoch 287/300
7352/7352 [==============================] - 35s 5ms/step - loss: 0.0993 - acc: 0.9584 - val_lo
Epoch 288/300
7352/7352 [==============================] - 34s 5ms/step - loss: 0.0944 - acc: 0.9603 - val_lo
Epoch 289/300
7352/7352 [==============================] - 42s 6ms/step - loss: 0.0933 - acc: 0.9581 - val_lo
Epoch 290/300
7352/7352 [==============================] - 36s 5ms/step - loss: 0.0993 - acc: 0.9574 - val_lo
Epoch 291/300
7352/7352 [==============================] - 35s 5ms/step - loss: 0.0943 - acc: 0.9585 - val_lo
Epoch 292/300
7352/7352 [==============================] - 34s 5ms/step - loss: 0.0960 - acc: 0.9593 - val_lo
Epoch 293/300
7352/7352 [==============================] - 34s 5ms/step - loss: 0.0958 - acc: 0.9597 - val_lo
Epoch 294/300
7352/7352 [==============================] - 33s 5ms/step - loss: 0.0945 - acc: 0.9576 - val_lo
Epoch 295/300
7352/7352 [==============================] - 34s 5ms/step - loss: 0.1000 - acc: 0.9573 - val_lo
Epoch 296/300
7352/7352 [==============================] - 32s 4ms/step - loss: 0.1204 - acc: 0.9567 - val_lo
Epoch 297/300
7352/7352 [==============================] - 33s 4ms/step - loss: 0.0881 - acc: 0.9604 - val_lo
Epoch 298/300
7352/7352 [==============================] - 34s 5ms/step - loss: 0.0848 - acc: 0.9618 - val_lo
Epoch 299/300
7352/7352 [==============================] - 33s 5ms/step - loss: 0.0939 - acc: 0.9603 - val_lo
Epoch 300/300
7352/7352 [==============================] - 32s 4ms/step - loss: 0.0990 - acc: 0.9582 - val_lo
```

Out[36]: <keras.callbacks.History at 0x1f1cd453860>

In [37]:
```
# Confusion Matrix
print(confusion_matrix(Y_test, model.predict(X_test)))
```

```
Pred                LAYING  SITTING  STANDING  WALKING  WALKING_DOWNSTAIRS  \
True
LAYING                 537        0         0        0                   0
SITTING                  0      378       113        0                   0
STANDING                 0       59       473        0                   0
WALKING                  0        0         0      496                   0
WALKING_DOWNSTAIRS       0        1         0       12                 399
WALKING_UPSTAIRS         0        2         1        3                   0

Pred                WALKING_UPSTAIRS
True
LAYING                             0
SITTING                            0
```

```
STANDING                       0
WALKING                        0
WALKING_DOWNSTAIRS             8
WALKING_UPSTAIRS             465
```

```
In [38]: score = model.evaluate(X_test, Y_test)
```

```
2947/2947 [==============================] - 3s 897us/step
```

```
In [39]: score
```

```
Out[39]: [0.21770981482614904, 0.9324737020699015]
```

- With a simple 2 layer architecture we got 93.24% accuracy and a loss of 0.21
- We can further imporve the performace with Hyperparameter tuning

17 Conclusion :

1.Here we seen thatby using two layered LSTM, we got a accuracy of 93.24% with loss of 0.21% .DeeP Learning help us to built models even when we don't have domain expert engineered features. 2.LSTM model can give good result by doing more hypertuning. 3.In this case study for better accuracy I had checked different epochs with different parameters.