CNN-LSTM-Self Attention

(i) Importing the necessary packages

In [ ]:

```
import numpy as np
import pandas as pd
import tensorflow as tf
import keras
from keras_self_attention import SeqSelfAttention

from keras import layers
from keras.models import Sequential,Model
from keras.layers import Dense, Activation, Flatten,Dropout, MultiHeadAttent
from keras.layers import Conv2D,LSTM,BatchNormalization,MaxPooling2D,Reshape
from keras.utils import to_categorical
import matplotlib.pyplot as plt
```

(ii) Preprocessing the dataset and preparing the training, validation, and test datasets

In [ ]:

```
def data_prep(X,y,sub_sample,average,noise):

    total_X = None
    total_y = None

    # Trimming the data (sample,22,1000) -> (sample,22,500)
    X = X[:,:,0:500]
    print('Shape of X after trimming:',X.shape)

    # Maxpooling the data (sample,22,1000) -> (sample,22,500/sub_sample)
    X_max = np.max(X.reshape(X.shape[0], X.shape[1], -1, sub_sample), axis=3


    total_X = X_max
    total_y = y
    print('Shape of X after maxpooling:',total_X.shape)

    # Averaging + noise
    X_average = np.mean(X.reshape(X.shape[0], X.shape[1], -1, average),axis=
    X_average = X_average + np.random.normal(0.0, 0.5, X_average.shape)

    total_X = np.vstack((total_X, X_average))
    total_y = np.hstack((total_y, y))
    print('Shape of X after averaging+noise and concatenating:',total_X.shap

    # Subsampling

    for i in range(sub_sample):

        X_subsample = X[:, :, i::sub_sample] + \
                            (np.random.normal(0.0, 0.5, X[:, :,i::sub_sample

        total_X = np.vstack((total_X, X_subsample))
        total_y = np.hstack((total_y, y))


    print('Shape of X after subsampling and concatenating:',total_X.shape)
    return total_X,total_y
```

In [ ]:

```
## Loading the dataset


X_test = np.load("X_test.npy")
y_test = np.load("y_test.npy")
person_train_valid = np.load("person_train_valid.npy")
X_train_valid = np.load("X_train_valid.npy")
y_train_valid = np.load("y_train_valid.npy")
person_test = np.load("person_test.npy")

## Adjusting the labels so that

# Cue onset left - 0
# Cue onset right - 1
# Cue onset foot - 2
# Cue onset tongue - 3

y_train_valid -= 769
y_test -= 769


## Random splitting and reshaping the data
# First generating the training and validation indices using random splittir

ind_valid = np.random.choice(2115, 375, replace=False)
ind_train = np.array(list(set(range(2115)).difference(set(ind_valid))))

# Creating the training and validation sets using the generated indices
(X_train, X_valid) = X_train_valid[ind_train], X_train_valid[ind_valid]
(y_train, y_valid) = y_train_valid[ind_train], y_train_valid[ind_valid]


## Preprocessing the dataset
x_train,y_train = data_prep(X_train,y_train,2,2,True)
x_valid,y_valid = data_prep(X_valid,y_valid,2,2,True)
X_test_prep,y_test_prep = data_prep(X_test,y_test,2,2,True)


print('Shape of training set:',x_train.shape)
print('Shape of validation set:',x_valid.shape)
print('Shape of training labels:',y_train.shape)
print('Shape of validation labels:',y_valid.shape)
print('Shape of testing set:',X_test_prep.shape)
print('Shape of testing labels:',y_test_prep.shape)


# Converting the labels to categorical variables for multiclass classificati
y_train = to_categorical(y_train, 4)
y_valid = to_categorical(y_valid, 4)
y_test = to_categorical(y_test_prep, 4)
print('Shape of training labels after categorical conversion:',y_train.shape
print('Shape of validation labels after categorical conversion:',y_valid.sha
print('Shape of test labels after categorical conversion:',y_test.shape)

# Adding width of the segment to be 1
x_train = x_train.reshape(x_train.shape[0], x_train.shape[1], x_train.shape[
x_valid = x_valid.reshape(x_valid.shape[0], x_valid.shape[1], x_train.shape[
x_test = X_test_prep.reshape(X_test_prep.shape[0], X_test_prep.shape[1], X_t
```

```
print('Shape of training set after adding width info:',x_train.shape)
print('Shape of validation set after adding width info:',x_valid.shape)
print('Shape of test set after adding width info:',x_test.shape)


# Reshaping the training and validation dataset
x_train = np.swapaxes(x_train, 1,3)
x_train = np.swapaxes(x_train, 1,2)
x_valid = np.swapaxes(x_valid, 1,3)
x_valid = np.swapaxes(x_valid, 1,2)
x_test = np.swapaxes(x_test, 1,3)
x_test = np.swapaxes(x_test, 1,2)
print('Shape of training set after dimension reshaping:',x_train.shape)
print('Shape of validation set after dimension reshaping:',x_valid.shape)
print('Shape of test set after dimension reshaping:',x_test.shape)

keras.backend.clear_session()
```

```
Shape of X after trimming: (1740, 22, 500)
Shape of X after maxpooling: (1740, 22, 250)
Shape of X after averaging+noise and concatenating: (3480, 22, 250)
Shape of X after subsampling and concatenating: (6960, 22, 250)
Shape of X after trimming: (375, 22, 500)
Shape of X after maxpooling: (375, 22, 250)
Shape of X after averaging+noise and concatenating: (750, 22, 250)
Shape of X after subsampling and concatenating: (1500, 22, 250)
Shape of X after trimming: (443, 22, 500)
Shape of X after maxpooling: (443, 22, 250)
Shape of X after averaging+noise and concatenating: (886, 22, 250)
Shape of X after subsampling and concatenating: (1772, 22, 250)
Shape of training set: (6960, 22, 250)
Shape of validation set: (1500, 22, 250)
Shape of training labels: (6960,)
Shape of validation labels: (1500,)
Shape of testing set: (1772, 22, 250)
Shape of testing labels: (1772,)
Shape of training labels after categorical conversion: (6960, 4)
Shape of validation labels after categorical conversion: (1500, 4)
Shape of test labels after categorical conversion: (1772, 4)
Shape of training set after adding width info: (6960, 22, 250, 1)
Shape of validation set after adding width info: (1500, 22, 250, 1)
Shape of test set after adding width info: (1772, 22, 250, 1)
Shape of training set after dimension reshaping: (6960, 250, 1, 22)
Shape of validation set after dimension reshaping: (1500, 250, 1, 22)
Shape of test set after dimension reshaping: (1772, 250, 1, 22)
```

(iii)(CNN-LSTM) Defining the architecture of the hybrid CNN-LSTM model

In [ ]:
```
def build_model(num_heads=2, input_shape=(250, 1, 22), num_classes=4):
    n_frames = 250
    n_channels = 22
    # Conv. block 1
    In1 = keras.Input(shape =(250,1,22) )
    c1 = Conv2D(filters=30, kernel_size=(11,1), padding='same', activation='
    p1 =  MaxPooling2D(pool_size=(4,1), padding='same')(c1) # Read the keras
    b1 = BatchNormalization()(p1)
    d1 = Dropout(0.6)(b1)

    # Conv. block 2
    c2 = Conv2D(filters=60, kernel_size=(9,1), padding='same', activation='s
    p2 =  MaxPooling2D(pool_size=(4,1), padding='same')(c2) # Read the keras
    b2 = BatchNormalization()(p2)
    d2 = Dropout(0.6)(b2)

    # Conv. block 3
    c3 = Conv2D(filters=120, kernel_size=(5,1), padding='same', activation='
    p3 =  MaxPooling2D(pool_size=(4,1), padding='same')(c3) # Read the keras
    b3 = BatchNormalization()(p3)
    d3 = Dropout(0.6)(b3)

    # Conv. block 4
    c4 = Conv2D(filters=240, kernel_size=(3,1), padding='same', activation='
    p4 =  MaxPooling2D(pool_size=(4,1), padding='same')(c4) # Read the keras
    b4 = BatchNormalization()(p4)
    d4 = Dropout(0.6)(b4)

    # LSTM block
    lstm1 = Reshape((1, -1))(d4)
    lstm1 = Bidirectional(LSTM(units=124, dropout=0.6, recurrent_dropout=0.1
    lstm1 = LayerNormalization()(lstm1)

    # self attention block
    selfatt = SeqSelfAttention(attention_activation='gelu')(lstm1)
    selfatt = GlobalAveragePooling1D()(selfatt)



    # # # Add fully connected layers
    fc2 = Dense(4, activation='softmax')(selfatt)

    # Define the final model
    final_model = Model(inputs=In1, outputs=[fc2])
    final_model.summary()
    return final_model
```

(iv)(CNN-LSTM) Defining the hyperparameters of the hybrid CNN-LSTM model

In [ ]:
```
# Model parameters

epochs = 200
initial_learning_rate = 1e-3
decay_steps = 1000
decay_rate = 0.99

lr_schedule = tf.keras.optimizers.schedules.ExponentialDecay(
    initial_learning_rate,
    decay_steps=decay_steps,
    decay_rate=decay_rate
)

optimizer = keras.optimizers.Adam(learning_rate=lr_schedule)
```

(v)(CNN-LSTM) Compiling, training and validating the model

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

# Training and validating the model
callback = keras.callbacks.EarlyStopping(monitor='val_accuracy', patience=20
hybrid_cnn_lstm_model_results = hybrid_cnn_lstm_model.fit(x_train,
                y_train,
                batch_size=32,
                epochs=epochs,
                validation_data=(x_valid, y_valid),
                verbose=True,
                callbacks = [callback])
```

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 250, 1, 22)]      0

 conv2d (Conv2D)             (None, 250, 1, 30)        7290

 max_pooling2d (MaxPooling2D  (None, 63, 1, 30)        0
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 250, 1, 22)]      0

 conv2d (Conv2D)             (None, 250, 1, 30)        7290

 max_pooling2d (MaxPooling2D  (None, 63, 1, 30)        0
 )

 batch_normalization (BatchN  (None, 63, 1, 30)        120
 ormalization)

 dropout (Dropout)           (None, 63, 1, 30)         0

 conv2d_1 (Conv2D)           (None, 63, 1, 60)         16260

 max_pooling2d_1 (MaxPooling  (None, 16, 1, 60)        0
 2D)

 batch_normalization_1 (Batc  (None, 16, 1, 60)        240
 hNormalization)

 dropout_1 (Dropout)         (None, 16, 1, 60)         0

 conv2d_2 (Conv2D)           (None, 16, 1, 120)        36120

 max_pooling2d_2 (MaxPooling  (None, 4, 1, 120)        0
 2D)

 batch_normalization_2 (Batc  (None, 4, 1, 120)        480
 hNormalization)

 dropout_2 (Dropout)         (None, 4, 1, 120)         0

 conv2d_3 (Conv2D)           (None, 4, 1, 240)         86640

 max_pooling2d_3 (MaxPooling  (None, 1, 1, 240)        0
 2D)

 batch_normalization_3 (Batc  (None, 1, 1, 240)        960
 hNormalization)

 dropout_3 (Dropout)         (None, 1, 1, 240)         0

 reshape (Reshape)           (None, 1, 240)            0

 bidirectional (Bidirectiona  (None, 1, 248)           362080
 l)
```

```
 layer_normalization (LayerN  (None, 1, 248)           496
 ormalization)

 seq_self_attention (SeqSelf  (None, 1, 248)           15937
 Attention)

 global_average_pooling1d (G  (None, 248)              0
 lobalAveragePooling1D)

 dense (Dense)              (None, 4)                 996

=================================================================
Total params: 527,619
Trainable params: 526,719
Non-trainable params: 900
_________________________________________________________________
Epoch 1/200
218/218 [==============================] - 9s 22ms/step - loss: 1.7706 - ac
curacy: 0.2533 - val_loss: 1.6084 - val_accuracy: 0.2507
Epoch 2/200
218/218 [==============================] - 4s 17ms/step - loss: 1.5303 - ac
curacy: 0.2592 - val_loss: 1.4583 - val_accuracy: 0.2720
Epoch 3/200
218/218 [==============================] - 4s 19ms/step - loss: 1.4488 - ac
curacy: 0.2662 - val_loss: 1.4491 - val_accuracy: 0.2760
Epoch 4/200
218/218 [==============================] - 4s 18ms/step - loss: 1.4170 - ac
curacy: 0.2721 - val_loss: 1.4806 - val_accuracy: 0.2813
Epoch 5/200
218/218 [==============================] - 4s 18ms/step - loss: 1.3968 - ac
curacy: 0.2792 - val_loss: 1.4810 - val_accuracy: 0.2893
Epoch 6/200
218/218 [==============================] - 4s 18ms/step - loss: 1.3848 - ac
curacy: 0.2932 - val_loss: 1.4235 - val_accuracy: 0.2787
Epoch 7/200
218/218 [==============================] - 4s 18ms/step - loss: 1.3796 - ac
curacy: 0.2996 - val_loss: 1.4273 - val_accuracy: 0.3060
Epoch 8/200
218/218 [==============================] - 4s 18ms/step - loss: 1.3748 - ac
curacy: 0.3050 - val_loss: 1.4436 - val_accuracy: 0.2933
Epoch 9/200
218/218 [==============================] - 4s 18ms/step - loss: 1.3648 - ac
curacy: 0.3172 - val_loss: 1.4524 - val_accuracy: 0.2707
Epoch 10/200
218/218 [==============================] - 4s 18ms/step - loss: 1.3563 - ac
curacy: 0.3251 - val_loss: 1.4445 - val_accuracy: 0.3100
Epoch 11/200
218/218 [==============================] - 4s 18ms/step - loss: 1.3384 - ac
curacy: 0.3402 - val_loss: 1.4544 - val_accuracy: 0.3447
Epoch 12/200
218/218 [==============================] - 4s 20ms/step - loss: 1.3167 - ac
curacy: 0.3626 - val_loss: 1.3783 - val_accuracy: 0.3607
Epoch 13/200
218/218 [==============================] - 4s 18ms/step - loss: 1.3033 - ac
curacy: 0.3750 - val_loss: 1.3561 - val_accuracy: 0.3747
Epoch 14/200
218/218 [==============================] - 4s 18ms/step - loss: 1.2848 - ac
curacy: 0.3884 - val_loss: 1.3375 - val_accuracy: 0.3953
Epoch 15/200
```

```
218/218 [==============================] - 4s 18ms/step - loss: 1.2729 - ac
curacy: 0.3964 - val_loss: 1.3498 - val_accuracy: 0.3960
Epoch 16/200
218/218 [==============================] - 4s 18ms/step - loss: 1.2651 - ac
curacy: 0.4082 - val_loss: 1.3246 - val_accuracy: 0.3893
Epoch 17/200
218/218 [==============================] - 4s 17ms/step - loss: 1.2564 - ac
curacy: 0.4152 - val_loss: 1.2874 - val_accuracy: 0.4287
Epoch 18/200
218/218 [==============================] - 4s 18ms/step - loss: 1.2459 - ac
curacy: 0.4234 - val_loss: 1.2568 - val_accuracy: 0.4593
Epoch 19/200
218/218 [==============================] - 4s 18ms/step - loss: 1.2374 - ac
curacy: 0.4290 - val_loss: 1.2368 - val_accuracy: 0.4460
Epoch 20/200
218/218 [==============================] - 4s 18ms/step - loss: 1.2355 - ac
curacy: 0.4297 - val_loss: 1.2585 - val_accuracy: 0.4627
Epoch 21/200
218/218 [==============================] - 4s 18ms/step - loss: 1.2108 - ac
curacy: 0.4519 - val_loss: 1.2341 - val_accuracy: 0.4827
Epoch 22/200
218/218 [==============================] - 5s 22ms/step - loss: 1.2057 - ac
curacy: 0.4470 - val_loss: 1.2907 - val_accuracy: 0.4533
Epoch 23/200
218/218 [==============================] - 4s 18ms/step - loss: 1.2003 - ac
curacy: 0.4507 - val_loss: 1.2393 - val_accuracy: 0.4427
Epoch 24/200
218/218 [==============================] - 4s 19ms/step - loss: 1.1898 - ac
curacy: 0.4559 - val_loss: 1.2273 - val_accuracy: 0.4613
Epoch 25/200
218/218 [==============================] - 4s 18ms/step - loss: 1.1877 - ac
curacy: 0.4647 - val_loss: 1.2187 - val_accuracy: 0.4620
Epoch 26/200
218/218 [==============================] - 4s 18ms/step - loss: 1.1817 - ac
curacy: 0.4677 - val_loss: 1.1949 - val_accuracy: 0.4760
Epoch 27/200
218/218 [==============================] - 4s 18ms/step - loss: 1.1743 - ac
curacy: 0.4737 - val_loss: 1.1762 - val_accuracy: 0.4813
Epoch 28/200
218/218 [==============================] - 4s 16ms/step - loss: 1.1713 - ac
curacy: 0.4787 - val_loss: 1.1786 - val_accuracy: 0.4980
Epoch 29/200
218/218 [==============================] - 3s 15ms/step - loss: 1.1564 - ac
curacy: 0.4802 - val_loss: 1.2041 - val_accuracy: 0.5013
Epoch 30/200
218/218 [==============================] - 3s 13ms/step - loss: 1.1465 - ac
curacy: 0.4921 - val_loss: 1.1558 - val_accuracy: 0.5120
Epoch 31/200
218/218 [==============================] - 3s 14ms/step - loss: 1.1426 - ac
curacy: 0.5000 - val_loss: 1.1332 - val_accuracy: 0.5220
Epoch 32/200
218/218 [==============================] - 3s 16ms/step - loss: 1.1304 - ac
curacy: 0.5116 - val_loss: 1.1288 - val_accuracy: 0.5387
Epoch 33/200
218/218 [==============================] - 3s 14ms/step - loss: 1.1169 - ac
curacy: 0.5194 - val_loss: 1.1211 - val_accuracy: 0.5307
Epoch 34/200
218/218 [==============================] - 3s 13ms/step - loss: 1.1120 - ac
curacy: 0.5198 - val_loss: 1.1439 - val_accuracy: 0.5313
```

```
Epoch 35/200
218/218 [==============================] - 3s 14ms/step - loss: 1.0922 - ac
curacy: 0.5435 - val_loss: 1.1107 - val_accuracy: 0.5373
Epoch 36/200
218/218 [==============================] - 3s 14ms/step - loss: 1.0777 - ac
curacy: 0.5448 - val_loss: 1.0957 - val_accuracy: 0.5593
Epoch 37/200
218/218 [==============================] - 3s 14ms/step - loss: 1.0892 - ac
curacy: 0.5398 - val_loss: 1.0651 - val_accuracy: 0.5800
Epoch 38/200
218/218 [==============================] - 3s 14ms/step - loss: 1.0583 - ac
curacy: 0.5591 - val_loss: 1.0853 - val_accuracy: 0.5760
Epoch 39/200
218/218 [==============================] - 3s 14ms/step - loss: 1.0536 - ac
curacy: 0.5664 - val_loss: 1.0426 - val_accuracy: 0.6007
Epoch 40/200
218/218 [==============================] - 3s 14ms/step - loss: 1.0397 - ac
curacy: 0.5717 - val_loss: 1.0266 - val_accuracy: 0.6087
Epoch 41/200
218/218 [==============================] - 3s 15ms/step - loss: 1.0346 - ac
curacy: 0.5767 - val_loss: 1.0194 - val_accuracy: 0.6107
Epoch 42/200
218/218 [==============================] - 3s 14ms/step - loss: 1.0192 - ac
curacy: 0.5774 - val_loss: 1.0060 - val_accuracy: 0.6140
Epoch 43/200
218/218 [==============================] - 3s 14ms/step - loss: 1.0003 - ac
curacy: 0.5955 - val_loss: 0.9844 - val_accuracy: 0.6293
Epoch 44/200
218/218 [==============================] - 3s 15ms/step - loss: 0.9918 - ac
curacy: 0.5963 - val_loss: 0.9557 - val_accuracy: 0.6273
Epoch 45/200
218/218 [==============================] - 3s 15ms/step - loss: 0.9783 - ac
curacy: 0.6065 - val_loss: 0.9676 - val_accuracy: 0.6147
Epoch 46/200
218/218 [==============================] - 3s 14ms/step - loss: 0.9884 - ac
curacy: 0.5996 - val_loss: 0.9613 - val_accuracy: 0.6267
Epoch 47/200
218/218 [==============================] - 3s 14ms/step - loss: 0.9546 - ac
curacy: 0.6121 - val_loss: 0.9217 - val_accuracy: 0.6333
Epoch 48/200
218/218 [==============================] - 3s 13ms/step - loss: 0.9514 - ac
curacy: 0.6144 - val_loss: 0.9535 - val_accuracy: 0.6273
Epoch 49/200
218/218 [==============================] - 3s 14ms/step - loss: 0.9567 - ac
curacy: 0.6141 - val_loss: 0.9620 - val_accuracy: 0.6433
Epoch 50/200
218/218 [==============================] - 3s 14ms/step - loss: 0.9487 - ac
curacy: 0.6201 - val_loss: 0.9379 - val_accuracy: 0.6500
Epoch 51/200
218/218 [==============================] - 3s 15ms/step - loss: 0.9380 - ac
curacy: 0.6250 - val_loss: 0.9080 - val_accuracy: 0.6587
Epoch 52/200
218/218 [==============================] - 3s 14ms/step - loss: 0.9305 - ac
curacy: 0.6276 - val_loss: 0.9093 - val_accuracy: 0.6547
Epoch 53/200
218/218 [==============================] - 3s 14ms/step - loss: 0.9114 - ac
curacy: 0.6445 - val_loss: 0.8866 - val_accuracy: 0.6620
Epoch 54/200
218/218 [==============================] - 3s 13ms/step - loss: 0.9074 - ac
```

```
curacy: 0.6409 - val_loss: 0.8938 - val_accuracy: 0.6500
Epoch 55/200
218/218 [==============================] - 3s 14ms/step - loss: 0.9141 - ac
curacy: 0.6368 - val_loss: 0.8879 - val_accuracy: 0.6613
Epoch 56/200
218/218 [==============================] - 3s 13ms/step - loss: 0.8951 - ac
curacy: 0.6438 - val_loss: 0.9048 - val_accuracy: 0.6587
Epoch 57/200
218/218 [==============================] - 3s 14ms/step - loss: 0.8966 - ac
curacy: 0.6372 - val_loss: 0.9343 - val_accuracy: 0.6500
Epoch 58/200
218/218 [==============================] - 3s 14ms/step - loss: 0.8995 - ac
curacy: 0.6366 - val_loss: 0.8736 - val_accuracy: 0.6533
Epoch 59/200
218/218 [==============================] - 3s 14ms/step - loss: 0.8881 - ac
curacy: 0.6441 - val_loss: 0.9022 - val_accuracy: 0.6600
Epoch 60/200
218/218 [==============================] - 3s 14ms/step - loss: 0.8665 - ac
curacy: 0.6579 - val_loss: 0.8582 - val_accuracy: 0.6847
Epoch 61/200
218/218 [==============================] - 3s 14ms/step - loss: 0.8731 - ac
curacy: 0.6540 - val_loss: 0.8959 - val_accuracy: 0.6600
Epoch 62/200
218/218 [==============================] - 3s 13ms/step - loss: 0.8737 - ac
curacy: 0.6506 - val_loss: 0.8673 - val_accuracy: 0.6600
Epoch 63/200
218/218 [==============================] - 3s 14ms/step - loss: 0.8636 - ac
curacy: 0.6562 - val_loss: 0.8622 - val_accuracy: 0.6620
Epoch 64/200
218/218 [==============================] - 3s 13ms/step - loss: 0.8537 - ac
curacy: 0.6667 - val_loss: 0.8862 - val_accuracy: 0.6413
Epoch 65/200
218/218 [==============================] - 3s 13ms/step - loss: 0.8565 - ac
curacy: 0.6631 - val_loss: 0.8682 - val_accuracy: 0.6793
Epoch 66/200
218/218 [==============================] - 3s 14ms/step - loss: 0.8451 - ac
curacy: 0.6639 - val_loss: 0.8812 - val_accuracy: 0.6740
Epoch 67/200
218/218 [==============================] - 3s 14ms/step - loss: 0.8554 - ac
curacy: 0.6736 - val_loss: 0.8109 - val_accuracy: 0.7000
Epoch 68/200
218/218 [==============================] - 3s 14ms/step - loss: 0.8365 - ac
curacy: 0.6698 - val_loss: 0.8635 - val_accuracy: 0.6813
Epoch 69/200
218/218 [==============================] - 3s 13ms/step - loss: 0.8481 - ac
curacy: 0.6674 - val_loss: 0.8491 - val_accuracy: 0.6893
Epoch 70/200
218/218 [==============================] - 3s 13ms/step - loss: 0.8340 - ac
curacy: 0.6744 - val_loss: 0.8419 - val_accuracy: 0.6853
Epoch 71/200
218/218 [==============================] - 3s 14ms/step - loss: 0.8234 - ac
curacy: 0.6705 - val_loss: 0.8427 - val_accuracy: 0.6913
Epoch 72/200
218/218 [==============================] - 3s 13ms/step - loss: 0.8252 - ac
curacy: 0.6815 - val_loss: 0.8506 - val_accuracy: 0.6807
Epoch 73/200
218/218 [==============================] - 3s 13ms/step - loss: 0.8208 - ac
curacy: 0.6750 - val_loss: 0.8613 - val_accuracy: 0.6807
Epoch 74/200
```

```
218/218 [==============================] - 3s 15ms/step - loss: 0.8243 - ac
curacy: 0.6777 - val_loss: 0.8547 - val_accuracy: 0.6880
Epoch 75/200
218/218 [==============================] - 3s 13ms/step - loss: 0.8109 - ac
curacy: 0.6829 - val_loss: 0.8944 - val_accuracy: 0.6713
Epoch 76/200
218/218 [==============================] - 3s 13ms/step - loss: 0.8064 - ac
curacy: 0.6846 - val_loss: 0.8365 - val_accuracy: 0.6853
Epoch 77/200
218/218 [==============================] - 3s 13ms/step - loss: 0.8044 - ac
curacy: 0.6861 - val_loss: 0.8286 - val_accuracy: 0.6887
Epoch 78/200
218/218 [==============================] - 3s 14ms/step - loss: 0.7959 - ac
curacy: 0.6924 - val_loss: 0.8424 - val_accuracy: 0.6953
Epoch 79/200
218/218 [==============================] - 3s 14ms/step - loss: 0.7945 - ac
curacy: 0.6865 - val_loss: 0.8245 - val_accuracy: 0.6880
Epoch 80/200
218/218 [==============================] - 4s 16ms/step - loss: 0.7956 - ac
curacy: 0.6908 - val_loss: 0.8079 - val_accuracy: 0.6900
Epoch 81/200
218/218 [==============================] - 4s 19ms/step - loss: 0.7771 - ac
curacy: 0.6945 - val_loss: 0.8684 - val_accuracy: 0.6933
Epoch 82/200
218/218 [==============================] - 5s 22ms/step - loss: 0.7828 - ac
curacy: 0.6977 - val_loss: 0.8150 - val_accuracy: 0.6933
Epoch 83/200
218/218 [==============================] - 5s 22ms/step - loss: 0.7708 - ac
curacy: 0.7020 - val_loss: 0.8339 - val_accuracy: 0.6787
Epoch 84/200
218/218 [==============================] - 5s 22ms/step - loss: 0.7928 - ac
curacy: 0.6898 - val_loss: 0.7983 - val_accuracy: 0.6987
Epoch 85/200
218/218 [==============================] - 5s 22ms/step - loss: 0.7898 - ac
curacy: 0.6898 - val_loss: 0.8378 - val_accuracy: 0.6860
Epoch 86/200
218/218 [==============================] - 4s 20ms/step - loss: 0.7696 - ac
curacy: 0.7019 - val_loss: 0.8059 - val_accuracy: 0.6960
Epoch 87/200
218/218 [==============================] - 5s 24ms/step - loss: 0.7814 - ac
curacy: 0.6999 - val_loss: 0.8118 - val_accuracy: 0.7027
Epoch 88/200
218/218 [==============================] - 5s 23ms/step - loss: 0.7791 - ac
curacy: 0.7009 - val_loss: 0.8154 - val_accuracy: 0.6973
Epoch 89/200
218/218 [==============================] - 5s 23ms/step - loss: 0.7644 - ac
curacy: 0.7039 - val_loss: 0.8034 - val_accuracy: 0.6993
Epoch 90/200
218/218 [==============================] - 5s 22ms/step - loss: 0.7572 - ac
curacy: 0.7039 - val_loss: 0.8146 - val_accuracy: 0.6900
Epoch 91/200
218/218 [==============================] - 5s 21ms/step - loss: 0.7623 - ac
curacy: 0.7068 - val_loss: 0.8115 - val_accuracy: 0.6987
Epoch 92/200
218/218 [==============================] - 5s 23ms/step - loss: 0.7484 - ac
curacy: 0.7102 - val_loss: 0.8236 - val_accuracy: 0.6880
Epoch 93/200
218/218 [==============================] - 5s 22ms/step - loss: 0.7496 - ac
curacy: 0.7103 - val_loss: 0.8296 - val_accuracy: 0.6887
```

```
Epoch 94/200
218/218 [==============================] - 5s 22ms/step - loss: 0.7564 - ac
curacy: 0.7132 - val_loss: 0.8102 - val_accuracy: 0.6940
Epoch 95/200
218/218 [==============================] - 5s 22ms/step - loss: 0.7544 - ac
curacy: 0.7121 - val_loss: 0.7939 - val_accuracy: 0.7000
Epoch 96/200
218/218 [==============================] - 5s 22ms/step - loss: 0.7502 - ac
curacy: 0.7106 - val_loss: 0.8095 - val_accuracy: 0.7000
Epoch 97/200
218/218 [==============================] - 5s 21ms/step - loss: 0.7566 - ac
curacy: 0.7063 - val_loss: 0.7860 - val_accuracy: 0.6973
Epoch 98/200
218/218 [==============================] - 5s 23ms/step - loss: 0.7453 - ac
curacy: 0.7111 - val_loss: 0.7874 - val_accuracy: 0.7127
Epoch 99/200
218/218 [==============================] - 4s 21ms/step - loss: 0.7433 - ac
curacy: 0.7116 - val_loss: 0.8281 - val_accuracy: 0.6913
Epoch 100/200
218/218 [==============================] - 4s 21ms/step - loss: 0.7355 - ac
curacy: 0.7132 - val_loss: 0.8037 - val_accuracy: 0.7027
Epoch 101/200
218/218 [==============================] - 5s 23ms/step - loss: 0.7364 - ac
curacy: 0.7195 - val_loss: 0.7993 - val_accuracy: 0.7040
Epoch 102/200
218/218 [==============================] - 5s 22ms/step - loss: 0.7388 - ac
curacy: 0.7154 - val_loss: 0.8082 - val_accuracy: 0.7127
Epoch 103/200
218/218 [==============================] - 5s 21ms/step - loss: 0.7334 - ac
curacy: 0.7165 - val_loss: 0.8104 - val_accuracy: 0.7100
Epoch 104/200
218/218 [==============================] - 5s 23ms/step - loss: 0.7168 - ac
curacy: 0.7228 - val_loss: 0.8311 - val_accuracy: 0.6907
Epoch 105/200
218/218 [==============================] - 5s 21ms/step - loss: 0.7298 - ac
curacy: 0.7185 - val_loss: 0.8098 - val_accuracy: 0.6973
Epoch 106/200
218/218 [==============================] - 5s 22ms/step - loss: 0.7224 - ac
curacy: 0.7216 - val_loss: 0.8183 - val_accuracy: 0.7073
Epoch 107/200
218/218 [==============================] - 5s 23ms/step - loss: 0.7218 - ac
curacy: 0.7198 - val_loss: 0.8175 - val_accuracy: 0.6980
Epoch 108/200
218/218 [==============================] - 5s 21ms/step - loss: 0.7197 - ac
curacy: 0.7240 - val_loss: 0.8019 - val_accuracy: 0.7007
Epoch 109/200
218/218 [==============================] - 5s 21ms/step - loss: 0.7263 - ac
curacy: 0.7227 - val_loss: 0.8067 - val_accuracy: 0.6940
Epoch 110/200
218/218 [==============================] - 5s 21ms/step - loss: 0.7129 - ac
curacy: 0.7213 - val_loss: 0.8316 - val_accuracy: 0.6980
Epoch 111/200
218/218 [==============================] - 5s 21ms/step - loss: 0.7169 - ac
curacy: 0.7244 - val_loss: 0.8222 - val_accuracy: 0.7067
Epoch 112/200
218/218 [==============================] - 5s 23ms/step - loss: 0.7152 - ac
curacy: 0.7236 - val_loss: 0.7948 - val_accuracy: 0.7093
Epoch 113/200
218/218 [==============================] - 4s 21ms/step - loss: 0.7264 - ac
```

```
curacy: 0.7175 - val_loss: 0.7847 - val_accuracy: 0.7100
Epoch 114/200
218/218 [==============================] - 5s 22ms/step - loss: 0.7024 - ac
curacy: 0.7292 - val_loss: 0.7891 - val_accuracy: 0.7140
Epoch 115/200
218/218 [==============================] - 5s 22ms/step - loss: 0.7252 - ac
curacy: 0.7249 - val_loss: 0.8450 - val_accuracy: 0.7013
Epoch 116/200
218/218 [==============================] - 5s 22ms/step - loss: 0.7023 - ac
curacy: 0.7251 - val_loss: 0.8033 - val_accuracy: 0.7153
Epoch 117/200
218/218 [==============================] - 5s 23ms/step - loss: 0.7115 - ac
curacy: 0.7287 - val_loss: 0.7836 - val_accuracy: 0.6913
Epoch 118/200
218/218 [==============================] - 5s 23ms/step - loss: 0.7033 - ac
curacy: 0.7326 - val_loss: 0.7737 - val_accuracy: 0.7147
Epoch 119/200
218/218 [==============================] - 5s 22ms/step - loss: 0.7119 - ac
curacy: 0.7226 - val_loss: 0.7818 - val_accuracy: 0.7073
Epoch 120/200
218/218 [==============================] - 5s 23ms/step - loss: 0.7106 - ac
curacy: 0.7276 - val_loss: 0.7806 - val_accuracy: 0.7113
Epoch 121/200
218/218 [==============================] - 5s 23ms/step - loss: 0.7131 - ac
curacy: 0.7249 - val_loss: 0.7736 - val_accuracy: 0.7180
Epoch 122/200
218/218 [==============================] - 6s 25ms/step - loss: 0.6994 - ac
curacy: 0.7310 - val_loss: 0.8013 - val_accuracy: 0.7053
Epoch 123/200
218/218 [==============================] - 5s 24ms/step - loss: 0.7053 - ac
curacy: 0.7277 - val_loss: 0.7816 - val_accuracy: 0.7100
Epoch 124/200
218/218 [==============================] - 5s 23ms/step - loss: 0.6994 - ac
curacy: 0.7338 - val_loss: 0.7855 - val_accuracy: 0.7140
Epoch 125/200
218/218 [==============================] - 5s 25ms/step - loss: 0.6769 - ac
curacy: 0.7398 - val_loss: 0.8210 - val_accuracy: 0.7040
Epoch 126/200
218/218 [==============================] - 5s 22ms/step - loss: 0.6864 - ac
curacy: 0.7336 - val_loss: 0.8061 - val_accuracy: 0.7067
Epoch 127/200
218/218 [==============================] - 5s 22ms/step - loss: 0.6982 - ac
curacy: 0.7300 - val_loss: 0.7828 - val_accuracy: 0.7173
Epoch 128/200
218/218 [==============================] - 5s 23ms/step - loss: 0.6793 - ac
curacy: 0.7409 - val_loss: 0.7991 - val_accuracy: 0.7213
Epoch 129/200
218/218 [==============================] - 5s 23ms/step - loss: 0.6927 - ac
curacy: 0.7361 - val_loss: 0.8099 - val_accuracy: 0.7120
Epoch 130/200
218/218 [==============================] - 5s 23ms/step - loss: 0.6833 - ac
curacy: 0.7356 - val_loss: 0.7737 - val_accuracy: 0.7200
Epoch 131/200
218/218 [==============================] - 5s 23ms/step - loss: 0.6770 - ac
curacy: 0.7445 - val_loss: 0.8112 - val_accuracy: 0.7107
Epoch 132/200
218/218 [==============================] - 5s 23ms/step - loss: 0.6687 - ac
curacy: 0.7470 - val_loss: 0.7771 - val_accuracy: 0.7280
Epoch 133/200
```

```
218/218 [==============================] - 5s 24ms/step - loss: 0.6677 - ac
curacy: 0.7476 - val_loss: 0.7952 - val_accuracy: 0.7140
Epoch 134/200
218/218 [==============================] - 5s 22ms/step - loss: 0.6915 - ac
curacy: 0.7333 - val_loss: 0.7854 - val_accuracy: 0.7120
Epoch 135/200
218/218 [==============================] - 6s 27ms/step - loss: 0.6659 - ac
curacy: 0.7430 - val_loss: 0.7829 - val_accuracy: 0.7193
Epoch 136/200
218/218 [==============================] - 5s 23ms/step - loss: 0.6687 - ac
curacy: 0.7401 - val_loss: 0.7602 - val_accuracy: 0.7220
Epoch 137/200
218/218 [==============================] - 5s 23ms/step - loss: 0.6726 - ac
curacy: 0.7415 - val_loss: 0.7693 - val_accuracy: 0.7147
Epoch 138/200
218/218 [==============================] - 5s 24ms/step - loss: 0.6691 - ac
curacy: 0.7450 - val_loss: 0.7824 - val_accuracy: 0.7173
Epoch 139/200
218/218 [==============================] - 5s 21ms/step - loss: 0.6573 - ac
curacy: 0.7504 - val_loss: 0.7811 - val_accuracy: 0.7027
Epoch 140/200
218/218 [==============================] - 5s 21ms/step - loss: 0.6666 - ac
curacy: 0.7372 - val_loss: 0.7932 - val_accuracy: 0.7133
Epoch 141/200
218/218 [==============================] - 5s 21ms/step - loss: 0.6715 - ac
curacy: 0.7417 - val_loss: 0.7731 - val_accuracy: 0.7147
Epoch 142/200
218/218 [==============================] - 5s 23ms/step - loss: 0.6819 - ac
curacy: 0.7391 - val_loss: 0.7669 - val_accuracy: 0.7173
Epoch 143/200
218/218 [==============================] - 5s 22ms/step - loss: 0.6508 - ac
curacy: 0.7468 - val_loss: 0.8015 - val_accuracy: 0.7187
Epoch 144/200
218/218 [==============================] - 5s 22ms/step - loss: 0.6586 - ac
curacy: 0.7494 - val_loss: 0.7884 - val_accuracy: 0.7227
Epoch 145/200
218/218 [==============================] - 5s 22ms/step - loss: 0.6355 - ac
curacy: 0.7553 - val_loss: 0.7763 - val_accuracy: 0.7240
Epoch 146/200
218/218 [==============================] - 5s 22ms/step - loss: 0.6454 - ac
curacy: 0.7549 - val_loss: 0.7590 - val_accuracy: 0.7233
Epoch 147/200
218/218 [==============================] - 5s 23ms/step - loss: 0.6600 - ac
curacy: 0.7430 - val_loss: 0.8053 - val_accuracy: 0.7140
Epoch 148/200
218/218 [==============================] - 5s 22ms/step - loss: 0.6545 - ac
curacy: 0.7510 - val_loss: 0.7602 - val_accuracy: 0.7313
Epoch 149/200
218/218 [==============================] - 5s 24ms/step - loss: 0.6615 - ac
curacy: 0.7418 - val_loss: 0.7660 - val_accuracy: 0.7267
Epoch 150/200
218/218 [==============================] - 5s 23ms/step - loss: 0.6388 - ac
curacy: 0.7591 - val_loss: 0.7630 - val_accuracy: 0.7333
Epoch 151/200
218/218 [==============================] - 5s 22ms/step - loss: 0.6239 - ac
curacy: 0.7639 - val_loss: 0.8129 - val_accuracy: 0.7160
Epoch 152/200
218/218 [==============================] - 5s 23ms/step - loss: 0.6503 - ac
curacy: 0.7504 - val_loss: 0.7669 - val_accuracy: 0.7247
```

```
Epoch 153/200
218/218 [==============================] - 5s 24ms/step - loss: 0.6462 - ac
curacy: 0.7545 - val_loss: 0.7681 - val_accuracy: 0.7313
Epoch 154/200
218/218 [==============================] - 5s 23ms/step - loss: 0.6525 - ac
curacy: 0.7468 - val_loss: 0.7879 - val_accuracy: 0.7240
Epoch 155/200
218/218 [==============================] - 5s 25ms/step - loss: 0.6795 - ac
curacy: 0.7376 - val_loss: 0.7587 - val_accuracy: 0.7300
Epoch 156/200
218/218 [==============================] - 5s 23ms/step - loss: 0.6477 - ac
curacy: 0.7520 - val_loss: 0.7637 - val_accuracy: 0.7153
Epoch 157/200
218/218 [==============================] - 5s 23ms/step - loss: 0.6443 - ac
curacy: 0.7534 - val_loss: 0.7714 - val_accuracy: 0.7287
Epoch 158/200
218/218 [==============================] - 5s 23ms/step - loss: 0.6465 - ac
curacy: 0.7503 - val_loss: 0.7434 - val_accuracy: 0.7367
Epoch 159/200
218/218 [==============================] - 5s 22ms/step - loss: 0.6361 - ac
curacy: 0.7589 - val_loss: 0.7864 - val_accuracy: 0.7207
Epoch 160/200
218/218 [==============================] - 5s 22ms/step - loss: 0.6345 - ac
curacy: 0.7579 - val_loss: 0.7932 - val_accuracy: 0.7160
Epoch 161/200
218/218 [==============================] - 5s 24ms/step - loss: 0.6427 - ac
curacy: 0.7560 - val_loss: 0.7789 - val_accuracy: 0.7193
Epoch 162/200
218/218 [==============================] - 5s 23ms/step - loss: 0.6506 - ac
curacy: 0.7524 - val_loss: 0.7639 - val_accuracy: 0.7300
Epoch 163/200
218/218 [==============================] - 5s 24ms/step - loss: 0.6319 - ac
curacy: 0.7575 - val_loss: 0.7576 - val_accuracy: 0.7387
Epoch 164/200
218/218 [==============================] - 5s 22ms/step - loss: 0.6426 - ac
curacy: 0.7578 - val_loss: 0.7635 - val_accuracy: 0.7253
Epoch 165/200
218/218 [==============================] - 5s 23ms/step - loss: 0.6299 - ac
curacy: 0.7593 - val_loss: 0.7492 - val_accuracy: 0.7293
Epoch 166/200
218/218 [==============================] - 5s 23ms/step - loss: 0.6185 - ac
curacy: 0.7674 - val_loss: 0.7571 - val_accuracy: 0.7300
Epoch 167/200
218/218 [==============================] - 5s 22ms/step - loss: 0.6233 - ac
curacy: 0.7592 - val_loss: 0.7638 - val_accuracy: 0.7347
Epoch 168/200
218/218 [==============================] - 5s 23ms/step - loss: 0.6295 - ac
curacy: 0.7615 - val_loss: 0.7428 - val_accuracy: 0.7340
Epoch 169/200
218/218 [==============================] - 5s 23ms/step - loss: 0.6346 - ac
curacy: 0.7556 - val_loss: 0.7496 - val_accuracy: 0.7340
Epoch 170/200
218/218 [==============================] - 5s 22ms/step - loss: 0.6191 - ac
curacy: 0.7611 - val_loss: 0.7466 - val_accuracy: 0.7307
Epoch 171/200
218/218 [==============================] - 6s 27ms/step - loss: 0.6276 - ac
curacy: 0.7599 - val_loss: 0.7556 - val_accuracy: 0.7367
Epoch 172/200
218/218 [==============================] - 4s 20ms/step - loss: 0.6225 - ac
```

```
curacy: 0.7652 - val_loss: 0.7655 - val_accuracy: 0.7273
Epoch 173/200
218/218 [==============================] - 5s 21ms/step - loss: 0.6208 - ac
curacy: 0.7618 - val_loss: 0.7604 - val_accuracy: 0.7347
Epoch 174/200
218/218 [==============================] - 5s 22ms/step - loss: 0.6252 - ac
curacy: 0.7612 - val_loss: 0.7347 - val_accuracy: 0.7360
Epoch 175/200
218/218 [==============================] - 5s 23ms/step - loss: 0.6155 - ac
curacy: 0.7619 - val_loss: 0.7463 - val_accuracy: 0.7333
Epoch 176/200
218/218 [==============================] - 5s 23ms/step - loss: 0.6190 - ac
curacy: 0.7622 - val_loss: 0.7883 - val_accuracy: 0.7213
Epoch 177/200
218/218 [==============================] - 5s 23ms/step - loss: 0.6355 - ac
curacy: 0.7560 - val_loss: 0.7409 - val_accuracy: 0.7313
Epoch 178/200
218/218 [==============================] - 5s 23ms/step - loss: 0.6140 - ac
curacy: 0.7672 - val_loss: 0.7677 - val_accuracy: 0.7380
Epoch 179/200
218/218 [==============================] - 5s 23ms/step - loss: 0.6157 - ac
curacy: 0.7645 - val_loss: 0.7584 - val_accuracy: 0.7253
Epoch 180/200
218/218 [==============================] - 5s 23ms/step - loss: 0.6325 - ac
curacy: 0.7565 - val_loss: 0.7561 - val_accuracy: 0.7300
Epoch 181/200
218/218 [==============================] - 5s 22ms/step - loss: 0.6063 - ac
curacy: 0.7668 - val_loss: 0.7545 - val_accuracy: 0.7360
Epoch 182/200
218/218 [==============================] - 5s 23ms/step - loss: 0.6061 - ac
curacy: 0.7726 - val_loss: 0.7674 - val_accuracy: 0.7320
Epoch 183/200
218/218 [==============================] - 5s 22ms/step - loss: 0.6124 - ac
curacy: 0.7661 - val_loss: 0.7575 - val_accuracy: 0.7320
```

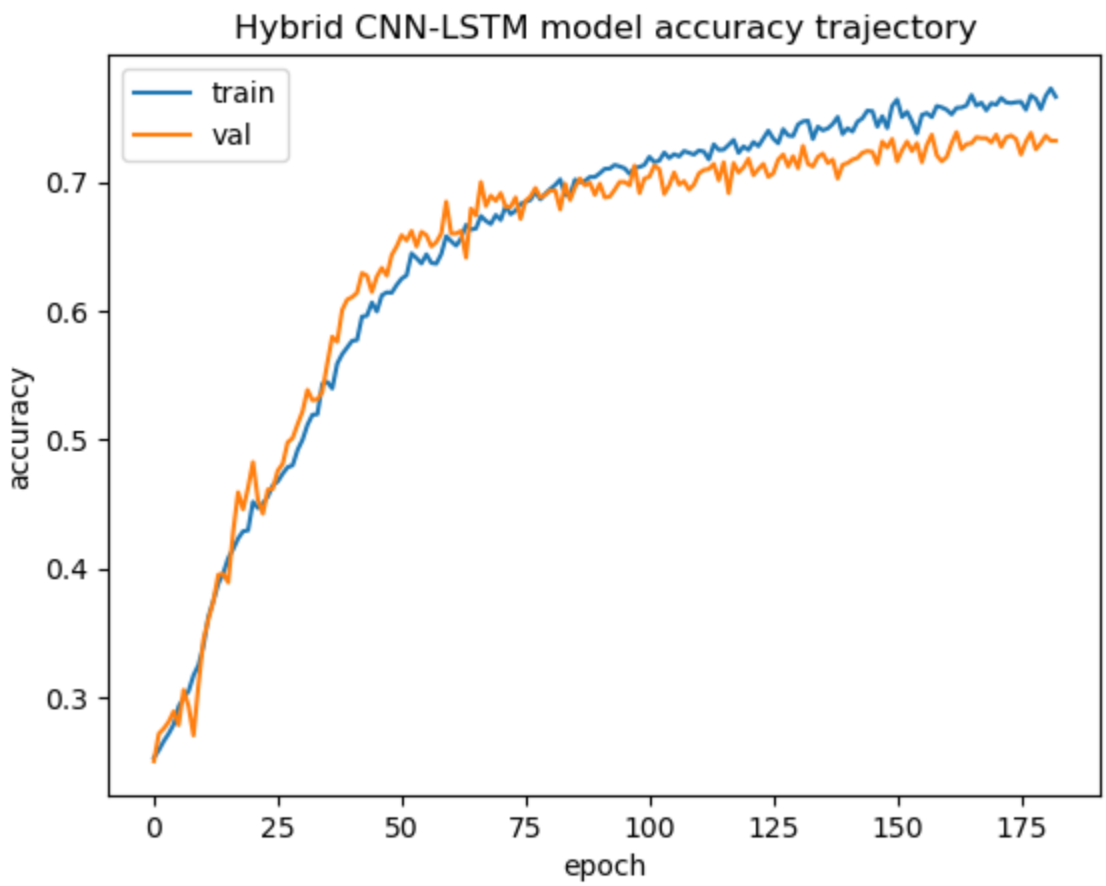
(vi)(CNN-LSTM) Visualizing the accuracy and loss trajectory

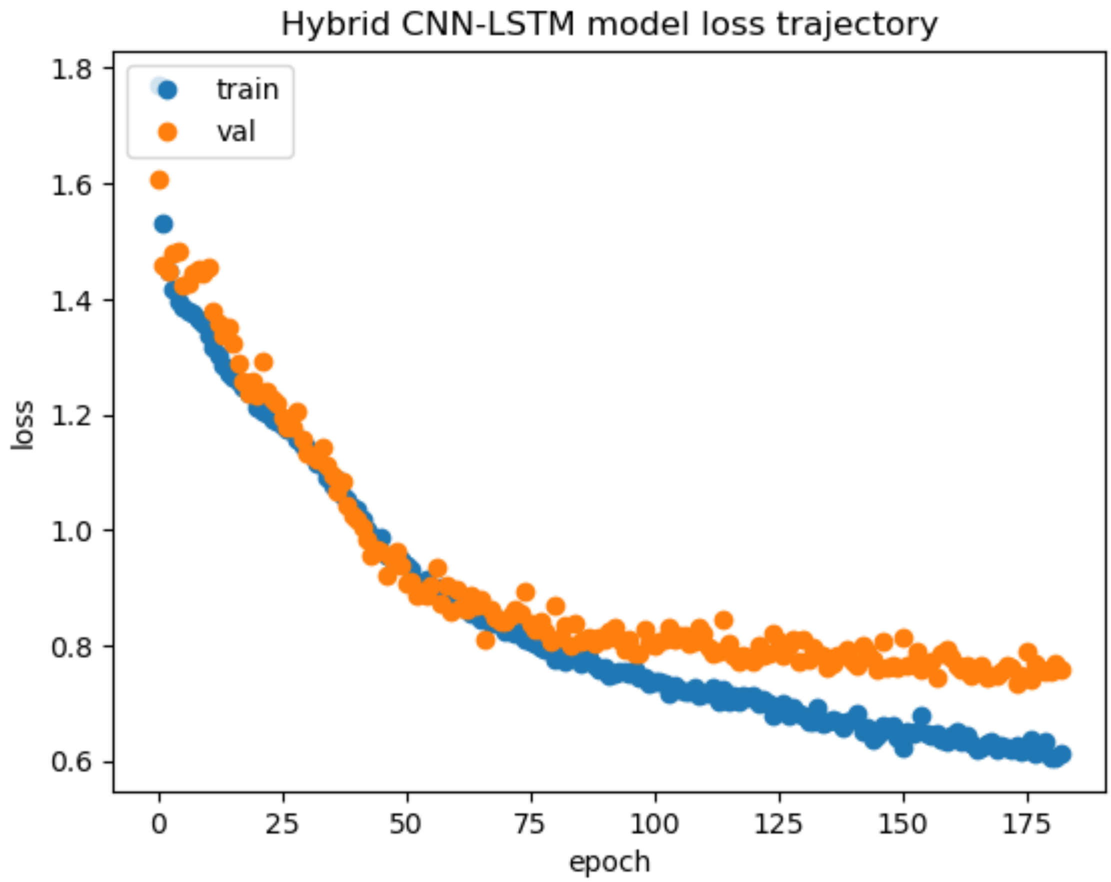
In [ ]:

```
import matplotlib.pyplot as plt

# Plotting accuracy trajectory
plt.plot(hybrid_cnn_lstm_model_results.history['accuracy'])
plt.plot(hybrid_cnn_lstm_model_results.history['val_accuracy'])
plt.title('Hybrid CNN-LSTM model accuracy trajectory')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()

# Plotting loss trajectory
plt.plot(hybrid_cnn_lstm_model_results.history['loss'],'o')
plt.plot(hybrid_cnn_lstm_model_results.history['val_loss'],'o')
plt.title('Hybrid CNN-LSTM model loss trajectory')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```





(vii)(CNN-LSTM) Testing the performance of the hybrid CNN-LSTM model on the held out test set

In [ ]:

```
## Testing the hybrid CNN-LSTM model

hybrid_cnn_lstm_score = hybrid_cnn_lstm_model.evaluate(x_test, y_test, verbo
print('Test accuracy of the hybrid CNN-LSTM-self_attention model:',hybrid_cn
```

```
Test accuracy of the hybrid CNN-LSTM-self_attention model: 0.70428895950317
38
```