In this discussion, we will build a basic hybrid CNN-MHA model for classification on the EEG dataset

This notebook was inspired to Tonmoy with some attempts to tune by us

(i) Importing the necessary packages

In [ ]:

```
import numpy as np
import pandas as pd
import keras
from keras import layers
from keras.models import Sequential, Model
from keras.layers import Dense, Activation, Flatten,Dropout,Bidirectional, G
from keras.layers import Conv2D, LayerNormalization, LSTM,BatchNormalizatior
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 [ ]:

```python
## 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 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-MHA) Defining the architecture of the hybrid CNN-MHA model

In [ ]:
```
# Building the CNN model using functional class
def build_model():
# models = []

    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(0.5)(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)


    #attention
    query_inputs = d4
    key_inputs = d4
    value_inputs = d4

    multi_head_attention = MultiHeadAttention(num_heads=8, key_dim=n_channel
    multi_head_attention = LayerNormalization(epsilon=1e-6)(multi_head_atter
    multi_head_attention = GlobalAveragePooling2D()(multi_head_attention)

    # Add fully connected layers
    fc1 = Dense(64, activation='relu')(multi_head_attention)
    fc1_dropout = Dropout(rate=0.5)(fc1)
    fc2 = Dense(4, activation='softmax')(fc1_dropout)

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

    # Compile the final model
    # # # Apply spatial attention
    # # hybrid_cnn_lstm_model.add(Lambda(spatial_attention))



    # # FC
    # hybrid_cnn_lstm_model.add(Flatten()) # Adding a flattening operation t
    # hybrid_cnn_lstm_model.add(Dense(100, activation='Elu')) # FC layer wit
```

```
    # hybrid_cnn_lstm_model.add(Reshape((100,1))) # Reshape my output of FC
    # # hybrid_cnn_lstm_model.add(LSTM(10, dropout=0.6, recurrent_dropout=0.


    # # Output layer with Softmax activation
    # hybrid_cnn_lstm_model.add(Dense(4, activation='softmax')) # Output FC

    # # models.append(hybrid_cnn_lstm_model)
    # # Printing the model summary
    final_model.compile(loss='categorical_crossentropy',
            optimizer='adam',
            metrics=['accuracy'])
    final_model.summary()
    return final_model
```

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

In [ ]:

```
import tensorflow as tf
# Model parameters
epochs = 300
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)Attention at the end

In [ ]:

```
# Compiling the model
keras.backend.clear_session()

# Training and validating the model

batch_sizes = [32,64,128]
import matplotlib.pyplot as plt

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
ACCURACY_THRESHOLD = 0.725
class myCallback(tf.keras.callbacks.Callback):
        def on_epoch_end(self, epoch, logs={}):
                if(logs.get('val_accuracy') > ACCURACY_THRESHOLD):
                        print("\nReached %2.2f%% accuracy, so stopping trair
                        self.model.stop_training = True

# Instantiate a callback object
callback = myCallback()
# callback = keras.callbacks.EarlyStopping(monitor='val_loss', patience=10)
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 #     Connected
to
==================================================================================================
 input_1 (InputLayer)           [(None, 250, 1, 22)  0           []
                                ]

 conv2d (Conv2D)                (None, 250, 1, 30)   7290        ['input_1
[0][0]']

 max_pooling2d (MaxPooling2D)   (None, 63, 1, 30)    0           ['conv2d
[0][0]']
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected
to
==================================================================================================
 input_1 (InputLayer)           [(None, 250, 1, 22)  0           []
                                ]

 conv2d (Conv2D)                (None, 250, 1, 30)   7290        ['input_1
[0][0]']

 max_pooling2d (MaxPooling2D)   (None, 63, 1, 30)    0           ['conv2d
[0][0]']

 batch_normalization (BatchNorm  (None, 63, 1, 30)   120         ['max_pool
ing2d[0][0]']
 alization)

 dropout (Dropout)              (None, 63, 1, 30)    0           ['batch_no
rmalization[0][0]']

 conv2d_1 (Conv2D)              (None, 63, 1, 60)    16260       ['dropout
[0][0]']

 max_pooling2d_1 (MaxPooling2D)  (None, 16, 1, 60)   0           ['conv2d_1
[0][0]']

 batch_normalization_1 (BatchNo  (None, 16, 1, 60)   240         ['max_pool
ing2d_1[0][0]']
 rmalization)

 dropout_1 (Dropout)            (None, 16, 1, 60)    0           ['batch_no
rmalization_1[0][0]']

 conv2d_2 (Conv2D)              (None, 16, 1, 120)   36120       ['dropout_
1[0][0]']

 max_pooling2d_2 (MaxPooling2D)  (None, 4, 1, 120)   0           ['conv2d_2
[0][0]']

 batch_normalization_2 (BatchNo  (None, 4, 1, 120)   480         ['max_pool
ing2d_2[0][0]']
 rmalization)
```

```
 dropout_2 (Dropout)            (None, 4, 1, 120)    0           ['batch_no
rmalization_2[0][0]']

 conv2d_3 (Conv2D)              (None, 4, 1, 240)    86640       ['dropout_
2[0][0]']

 max_pooling2d_3 (MaxPooling2D)  (None, 1, 1, 240)   0           ['conv2d_3
[0][0]']

 batch_normalization_3 (BatchNo  (None, 1, 1, 240)   960         ['max_pool
ing2d_3[0][0]']
 rmalization)

 dropout_3 (Dropout)            (None, 1, 1, 240)    0           ['batch_no
rmalization_3[0][0]']

 multi_head_attention (MultiHea  (None, 1, 1, 240)   15648       ['dropout_
3[0][0]',
 dAttention)                                                      'dropout_
3[0][0]',
                                                                  'dropout_

3[0][0]']

 layer_normalization (LayerNorm  (None, 1, 1, 240)   480         ['multi_he
ad_attention[0][0]']
 alization)

 global_average_pooling2d (Glob  (None, 240)         0           ['layer_no
rmalization[0][0]']
 alAveragePooling2D)

 dense (Dense)                  (None, 64)           15424       ['global_a
verage_pooling2d[0][0]'
                                                                 ]

 dropout_4 (Dropout)            (None, 64)           0           ['dense
[0][0]']

 dense_1 (Dense)                (None, 4)            260         ['dropout_
4[0][0]']

===========================================================================
=======================
Total params: 179,922
Trainable params: 179,022
Non-trainable params: 900
___________________________________________________________________________
_______________________
Epoch 1/300
218/218 [==============================] - 5s 18ms/step - loss: 1.5216 - ac
curacy: 0.2579 - val_loss: 1.3850 - val_accuracy: 0.2660
Epoch 2/300
218/218 [==============================] - 3s 16ms/step - loss: 1.3928 - ac
curacy: 0.2691 - val_loss: 1.3945 - val_accuracy: 0.2427
Epoch 3/300
218/218 [==============================] - 3s 16ms/step - loss: 1.3872 - ac
curacy: 0.2694 - val_loss: 1.3978 - val_accuracy: 0.2607
Epoch 4/300
```

```
218/218 [==============================] - 4s 17ms/step - loss: 1.3859 - ac
curacy: 0.2655 - val_loss: 1.3903 - val_accuracy: 0.2333
Epoch 5/300
218/218 [==============================] - 3s 16ms/step - loss: 1.3844 - ac
curacy: 0.2744 - val_loss: 1.3935 - val_accuracy: 0.2473
Epoch 6/300
218/218 [==============================] - 4s 17ms/step - loss: 1.3799 - ac
curacy: 0.2796 - val_loss: 1.3728 - val_accuracy: 0.3040
Epoch 7/300
218/218 [==============================] - 4s 17ms/step - loss: 1.3711 - ac
curacy: 0.2932 - val_loss: 1.3667 - val_accuracy: 0.2887
Epoch 8/300
218/218 [==============================] - 3s 16ms/step - loss: 1.3619 - ac
curacy: 0.3010 - val_loss: 1.3272 - val_accuracy: 0.3767
Epoch 9/300
218/218 [==============================] - 4s 16ms/step - loss: 1.3381 - ac
curacy: 0.3431 - val_loss: 1.2786 - val_accuracy: 0.4187
Epoch 10/300
218/218 [==============================] - 3s 16ms/step - loss: 1.3203 - ac
curacy: 0.3539 - val_loss: 1.2826 - val_accuracy: 0.4267
Epoch 11/300
218/218 [==============================] - 4s 17ms/step - loss: 1.3013 - ac
curacy: 0.3816 - val_loss: 1.2479 - val_accuracy: 0.4260
Epoch 12/300
218/218 [==============================] - 3s 16ms/step - loss: 1.2741 - ac
curacy: 0.4066 - val_loss: 1.2875 - val_accuracy: 0.3907
Epoch 13/300
218/218 [==============================] - 4s 16ms/step - loss: 1.2720 - ac
curacy: 0.4152 - val_loss: 1.2768 - val_accuracy: 0.3773
Epoch 14/300
218/218 [==============================] - 3s 16ms/step - loss: 1.2564 - ac
curacy: 0.4187 - val_loss: 1.2149 - val_accuracy: 0.4220
Epoch 15/300
218/218 [==============================] - 3s 15ms/step - loss: 1.2449 - ac
curacy: 0.4292 - val_loss: 1.1978 - val_accuracy: 0.4540
Epoch 16/300
218/218 [==============================] - 3s 13ms/step - loss: 1.2384 - ac
curacy: 0.4343 - val_loss: 1.2065 - val_accuracy: 0.4407
Epoch 17/300
218/218 [==============================] - 3s 12ms/step - loss: 1.2326 - ac
curacy: 0.4330 - val_loss: 1.1704 - val_accuracy: 0.4620
Epoch 18/300
218/218 [==============================] - 2s 11ms/step - loss: 1.2186 - ac
curacy: 0.4458 - val_loss: 1.1929 - val_accuracy: 0.4567
Epoch 19/300
218/218 [==============================] - 2s 11ms/step - loss: 1.2013 - ac
curacy: 0.4519 - val_loss: 1.1434 - val_accuracy: 0.4933
Epoch 20/300
218/218 [==============================] - 2s 11ms/step - loss: 1.1920 - ac
curacy: 0.4606 - val_loss: 1.1529 - val_accuracy: 0.4713
Epoch 21/300
218/218 [==============================] - 3s 12ms/step - loss: 1.1816 - ac
curacy: 0.4730 - val_loss: 1.2039 - val_accuracy: 0.4593
Epoch 22/300
218/218 [==============================] - 2s 11ms/step - loss: 1.1720 - ac
curacy: 0.4694 - val_loss: 1.1412 - val_accuracy: 0.4687
Epoch 23/300
218/218 [==============================] - 2s 11ms/step - loss: 1.1581 - ac
curacy: 0.4796 - val_loss: 1.1460 - val_accuracy: 0.4727
```

```
Epoch 24/300
218/218 [==============================] - 2s 11ms/step - loss: 1.1471 - ac
curacy: 0.4832 - val_loss: 1.1541 - val_accuracy: 0.4753
Epoch 25/300
218/218 [==============================] - 2s 11ms/step - loss: 1.1398 - ac
curacy: 0.4935 - val_loss: 1.1531 - val_accuracy: 0.4733
Epoch 26/300
218/218 [==============================] - 2s 11ms/step - loss: 1.1291 - ac
curacy: 0.4886 - val_loss: 1.1213 - val_accuracy: 0.4720
Epoch 27/300
218/218 [==============================] - 3s 12ms/step - loss: 1.1164 - ac
curacy: 0.4981 - val_loss: 1.1176 - val_accuracy: 0.4920
Epoch 28/300
218/218 [==============================] - 3s 12ms/step - loss: 1.1230 - ac
curacy: 0.4981 - val_loss: 1.1643 - val_accuracy: 0.4687
Epoch 29/300
218/218 [==============================] - 3s 12ms/step - loss: 1.1037 - ac
curacy: 0.5052 - val_loss: 1.1061 - val_accuracy: 0.4820
Epoch 30/300
218/218 [==============================] - 3s 12ms/step - loss: 1.0976 - ac
curacy: 0.5082 - val_loss: 1.1314 - val_accuracy: 0.4660
Epoch 31/300
218/218 [==============================] - 2s 11ms/step - loss: 1.0831 - ac
curacy: 0.5152 - val_loss: 1.1478 - val_accuracy: 0.4773
Epoch 32/300
218/218 [==============================] - 3s 12ms/step - loss: 1.0739 - ac
curacy: 0.5201 - val_loss: 1.0792 - val_accuracy: 0.4980
Epoch 33/300
218/218 [==============================] - 3s 13ms/step - loss: 1.0650 - ac
curacy: 0.5177 - val_loss: 1.1413 - val_accuracy: 0.4887
Epoch 34/300
218/218 [==============================] - 3s 12ms/step - loss: 1.0627 - ac
curacy: 0.5233 - val_loss: 1.1294 - val_accuracy: 0.4820
Epoch 35/300
218/218 [==============================] - 2s 11ms/step - loss: 1.0517 - ac
curacy: 0.5411 - val_loss: 1.1005 - val_accuracy: 0.4760
Epoch 36/300
218/218 [==============================] - 3s 11ms/step - loss: 1.0405 - ac
curacy: 0.5365 - val_loss: 1.1255 - val_accuracy: 0.4907
Epoch 37/300
218/218 [==============================] - 2s 11ms/step - loss: 1.0364 - ac
curacy: 0.5483 - val_loss: 1.0937 - val_accuracy: 0.5200
Epoch 38/300
218/218 [==============================] - 3s 12ms/step - loss: 1.0266 - ac
curacy: 0.5404 - val_loss: 1.1014 - val_accuracy: 0.5213
Epoch 39/300
218/218 [==============================] - 3s 12ms/step - loss: 1.0200 - ac
curacy: 0.5605 - val_loss: 1.0836 - val_accuracy: 0.5387
Epoch 40/300
218/218 [==============================] - 3s 12ms/step - loss: 1.0102 - ac
curacy: 0.5662 - val_loss: 1.0574 - val_accuracy: 0.5560
Epoch 41/300
218/218 [==============================] - 2s 11ms/step - loss: 1.0037 - ac
curacy: 0.5734 - val_loss: 1.0650 - val_accuracy: 0.5753
Epoch 42/300
218/218 [==============================] - 3s 12ms/step - loss: 0.9884 - ac
curacy: 0.5909 - val_loss: 1.0392 - val_accuracy: 0.5867
Epoch 43/300
218/218 [==============================] - 3s 12ms/step - loss: 0.9985 - ac
```

```
curacy: 0.5878 - val_loss: 1.0511 - val_accuracy: 0.5780
Epoch 44/300
218/218 [==============================] - 3s 12ms/step - loss: 0.9883 - ac
curacy: 0.5940 - val_loss: 1.0550 - val_accuracy: 0.5700
Epoch 45/300
218/218 [==============================] - 3s 12ms/step - loss: 0.9822 - ac
curacy: 0.5950 - val_loss: 1.0311 - val_accuracy: 0.5667
Epoch 46/300
218/218 [==============================] - 2s 11ms/step - loss: 0.9571 - ac
curacy: 0.6111 - val_loss: 0.9798 - val_accuracy: 0.6133
Epoch 47/300
218/218 [==============================] - 2s 11ms/step - loss: 0.9473 - ac
curacy: 0.6205 - val_loss: 0.9626 - val_accuracy: 0.6313
Epoch 48/300
218/218 [==============================] - 2s 11ms/step - loss: 0.9514 - ac
curacy: 0.6177 - val_loss: 0.9532 - val_accuracy: 0.6227
Epoch 49/300
218/218 [==============================] - 2s 11ms/step - loss: 0.9154 - ac
curacy: 0.6382 - val_loss: 0.9526 - val_accuracy: 0.6593
Epoch 50/300
218/218 [==============================] - 2s 11ms/step - loss: 0.9214 - ac
curacy: 0.6391 - val_loss: 0.9451 - val_accuracy: 0.6453
Epoch 51/300
218/218 [==============================] - 2s 11ms/step - loss: 0.9032 - ac
curacy: 0.6430 - val_loss: 0.9546 - val_accuracy: 0.6373
Epoch 52/300
218/218 [==============================] - 2s 11ms/step - loss: 0.8931 - ac
curacy: 0.6573 - val_loss: 0.9344 - val_accuracy: 0.6500
Epoch 53/300
218/218 [==============================] - 2s 11ms/step - loss: 0.8815 - ac
curacy: 0.6510 - val_loss: 0.8903 - val_accuracy: 0.6707
Epoch 54/300
218/218 [==============================] - 3s 11ms/step - loss: 0.8661 - ac
curacy: 0.6721 - val_loss: 0.9081 - val_accuracy: 0.6653
Epoch 55/300
218/218 [==============================] - 2s 11ms/step - loss: 0.8824 - ac
curacy: 0.6583 - val_loss: 0.8910 - val_accuracy: 0.6580
Epoch 56/300
218/218 [==============================] - 2s 11ms/step - loss: 0.8682 - ac
curacy: 0.6603 - val_loss: 0.8873 - val_accuracy: 0.6633
Epoch 57/300
218/218 [==============================] - 2s 11ms/step - loss: 0.8507 - ac
curacy: 0.6789 - val_loss: 0.9000 - val_accuracy: 0.6333
Epoch 58/300
218/218 [==============================] - 2s 11ms/step - loss: 0.8582 - ac
curacy: 0.6641 - val_loss: 0.8813 - val_accuracy: 0.6613
Epoch 59/300
218/218 [==============================] - 2s 11ms/step - loss: 0.8400 - ac
curacy: 0.6915 - val_loss: 0.8540 - val_accuracy: 0.6800
Epoch 60/300
218/218 [==============================] - 2s 11ms/step - loss: 0.8285 - ac
curacy: 0.6886 - val_loss: 0.8620 - val_accuracy: 0.6647
Epoch 61/300
218/218 [==============================] - 2s 11ms/step - loss: 0.8198 - ac
curacy: 0.6889 - val_loss: 0.8879 - val_accuracy: 0.6607
Epoch 62/300
218/218 [==============================] - 2s 11ms/step - loss: 0.8199 - ac
curacy: 0.6922 - val_loss: 0.8401 - val_accuracy: 0.6793
Epoch 63/300
```

```
218/218 [==============================] - 2s 11ms/step - loss: 0.8049 - ac
curacy: 0.6957 - val_loss: 0.8632 - val_accuracy: 0.6667
Epoch 64/300
218/218 [==============================] - 2s 11ms/step - loss: 0.8107 - ac
curacy: 0.6974 - val_loss: 0.8707 - val_accuracy: 0.6733
Epoch 65/300
218/218 [==============================] - 2s 11ms/step - loss: 0.8033 - ac
curacy: 0.6945 - val_loss: 0.8397 - val_accuracy: 0.6947
Epoch 66/300
218/218 [==============================] - 2s 11ms/step - loss: 0.8052 - ac
curacy: 0.7001 - val_loss: 0.8171 - val_accuracy: 0.7007
Epoch 67/300
218/218 [==============================] - 2s 11ms/step - loss: 0.7980 - ac
curacy: 0.6960 - val_loss: 0.8706 - val_accuracy: 0.6853
Epoch 68/300
218/218 [==============================] - 2s 11ms/step - loss: 0.7969 - ac
curacy: 0.7029 - val_loss: 0.8181 - val_accuracy: 0.7053
Epoch 69/300
218/218 [==============================] - 2s 11ms/step - loss: 0.7896 - ac
curacy: 0.7057 - val_loss: 0.8351 - val_accuracy: 0.6893
Epoch 70/300
218/218 [==============================] - 2s 11ms/step - loss: 0.7785 - ac
curacy: 0.7089 - val_loss: 0.8155 - val_accuracy: 0.6980
Epoch 71/300
218/218 [==============================] - 2s 11ms/step - loss: 0.7698 - ac
curacy: 0.7095 - val_loss: 0.8953 - val_accuracy: 0.6660
Epoch 72/300
218/218 [==============================] - 2s 11ms/step - loss: 0.7696 - ac
curacy: 0.7125 - val_loss: 0.8294 - val_accuracy: 0.6860
Epoch 73/300
218/218 [==============================] - 2s 11ms/step - loss: 0.7623 - ac
curacy: 0.7096 - val_loss: 0.8084 - val_accuracy: 0.6953
Epoch 74/300
218/218 [==============================] - 2s 11ms/step - loss: 0.7629 - ac
curacy: 0.7144 - val_loss: 0.8431 - val_accuracy: 0.6540
Epoch 75/300
218/218 [==============================] - 2s 11ms/step - loss: 0.7452 - ac
curacy: 0.7170 - val_loss: 0.7928 - val_accuracy: 0.6913
Epoch 76/300
218/218 [==============================] - 2s 11ms/step - loss: 0.7444 - ac
curacy: 0.7292 - val_loss: 0.8405 - val_accuracy: 0.6700
Epoch 77/300
218/218 [==============================] - 2s 11ms/step - loss: 0.7497 - ac
curacy: 0.7241 - val_loss: 0.8513 - val_accuracy: 0.6753
Epoch 78/300
218/218 [==============================] - 2s 11ms/step - loss: 0.7463 - ac
curacy: 0.7178 - val_loss: 0.8445 - val_accuracy: 0.6660
Epoch 79/300
218/218 [==============================] - 2s 11ms/step - loss: 0.7392 - ac
curacy: 0.7293 - val_loss: 0.8467 - val_accuracy: 0.6787
Epoch 80/300
218/218 [==============================] - 2s 11ms/step - loss: 0.7449 - ac
curacy: 0.7236 - val_loss: 0.8807 - val_accuracy: 0.6953
Epoch 81/300
218/218 [==============================] - 3s 12ms/step - loss: 0.7203 - ac
curacy: 0.7299 - val_loss: 0.8642 - val_accuracy: 0.6807
Epoch 82/300
218/218 [==============================] - 3s 12ms/step - loss: 0.7240 - ac
curacy: 0.7333 - val_loss: 0.8387 - val_accuracy: 0.6853
```

```
Epoch 83/300
218/218 [==============================] - 2s 11ms/step - loss: 0.7350 - ac
curacy: 0.7244 - val_loss: 0.8107 - val_accuracy: 0.6833
Epoch 84/300
218/218 [==============================] - 3s 12ms/step - loss: 0.7112 - ac
curacy: 0.7375 - val_loss: 0.8666 - val_accuracy: 0.6860
Epoch 85/300
218/218 [==============================] - 3s 13ms/step - loss: 0.7151 - ac
curacy: 0.7359 - val_loss: 0.8273 - val_accuracy: 0.6960
Epoch 86/300
218/218 [==============================] - 2s 11ms/step - loss: 0.7091 - ac
curacy: 0.7430 - val_loss: 0.7873 - val_accuracy: 0.7007
Epoch 87/300
218/218 [==============================] - 2s 11ms/step - loss: 0.7073 - ac
curacy: 0.7418 - val_loss: 0.8552 - val_accuracy: 0.6793
Epoch 88/300
218/218 [==============================] - 2s 11ms/step - loss: 0.7252 - ac
curacy: 0.7279 - val_loss: 0.8027 - val_accuracy: 0.7013
Epoch 89/300
218/218 [==============================] - 2s 11ms/step - loss: 0.7015 - ac
curacy: 0.7411 - val_loss: 0.8444 - val_accuracy: 0.7160
Epoch 90/300
218/218 [==============================] - 2s 11ms/step - loss: 0.6871 - ac
curacy: 0.7483 - val_loss: 0.8116 - val_accuracy: 0.6913
Epoch 91/300
218/218 [==============================] - 3s 12ms/step - loss: 0.7028 - ac
curacy: 0.7421 - val_loss: 0.7976 - val_accuracy: 0.7087
Epoch 92/300
218/218 [==============================] - 3s 12ms/step - loss: 0.7022 - ac
curacy: 0.7384 - val_loss: 0.7912 - val_accuracy: 0.7160
Epoch 93/300
218/218 [==============================] - 3s 12ms/step - loss: 0.7055 - ac
curacy: 0.7376 - val_loss: 0.8224 - val_accuracy: 0.6860
Epoch 94/300
218/218 [==============================] - 3s 12ms/step - loss: 0.6770 - ac
curacy: 0.7527 - val_loss: 0.8235 - val_accuracy: 0.6893
Epoch 95/300
218/218 [==============================] - 3s 12ms/step - loss: 0.6906 - ac
curacy: 0.7420 - val_loss: 0.7930 - val_accuracy: 0.7067
Epoch 96/300
218/218 [==============================] - 2s 11ms/step - loss: 0.6849 - ac
curacy: 0.7474 - val_loss: 0.7904 - val_accuracy: 0.6933
Epoch 97/300
218/218 [==============================] - 2s 11ms/step - loss: 0.6669 - ac
curacy: 0.7534 - val_loss: 0.8639 - val_accuracy: 0.6847
Epoch 98/300
218/218 [==============================] - 2s 11ms/step - loss: 0.6780 - ac
curacy: 0.7513 - val_loss: 0.8125 - val_accuracy: 0.6820
Epoch 99/300
218/218 [==============================] - 3s 12ms/step - loss: 0.6855 - ac
curacy: 0.7499 - val_loss: 0.7995 - val_accuracy: 0.7080
Epoch 100/300
218/218 [==============================] - 3s 12ms/step - loss: 0.6595 - ac
curacy: 0.7572 - val_loss: 0.8387 - val_accuracy: 0.7133
Epoch 101/300
218/218 [==============================] - 3s 12ms/step - loss: 0.6813 - ac
curacy: 0.7466 - val_loss: 0.8080 - val_accuracy: 0.7067
Epoch 102/300
218/218 [==============================] - 3s 12ms/step - loss: 0.6633 - ac
```

```
curacy: 0.7580 - val_loss: 0.7842 - val_accuracy: 0.7120
Epoch 103/300
218/218 [==============================] - 3s 12ms/step - loss: 0.6560 - ac
curacy: 0.7622 - val_loss: 0.7990 - val_accuracy: 0.6907
Epoch 104/300
218/218 [==============================] - 3s 13ms/step - loss: 0.6758 - ac
curacy: 0.7536 - val_loss: 0.8567 - val_accuracy: 0.6960
Epoch 105/300
218/218 [==============================] - 4s 17ms/step - loss: 0.6693 - ac
curacy: 0.7483 - val_loss: 0.8337 - val_accuracy: 0.7073
Epoch 106/300
218/218 [==============================] - 3s 14ms/step - loss: 0.6604 - ac
curacy: 0.7621 - val_loss: 0.8186 - val_accuracy: 0.7067
Epoch 107/300
218/218 [==============================] - 3s 16ms/step - loss: 0.6480 - ac
curacy: 0.7605 - val_loss: 0.7851 - val_accuracy: 0.7087
Epoch 108/300
218/218 [==============================] - 3s 15ms/step - loss: 0.6491 - ac
curacy: 0.7616 - val_loss: 0.8051 - val_accuracy: 0.6973
Epoch 109/300
218/218 [==============================] - 3s 16ms/step - loss: 0.6629 - ac
curacy: 0.7589 - val_loss: 0.7961 - val_accuracy: 0.7080
Epoch 110/300
218/218 [==============================] - 3s 15ms/step - loss: 0.6476 - ac
curacy: 0.7583 - val_loss: 0.8239 - val_accuracy: 0.6860
Epoch 111/300
218/218 [==============================] - 4s 16ms/step - loss: 0.6393 - ac
curacy: 0.7642 - val_loss: 0.8211 - val_accuracy: 0.6860
Epoch 112/300
218/218 [==============================] - 5s 24ms/step - loss: 0.6437 - ac
curacy: 0.7652 - val_loss: 0.8326 - val_accuracy: 0.6907
Epoch 113/300
218/218 [==============================] - 4s 17ms/step - loss: 0.6440 - ac
curacy: 0.7667 - val_loss: 0.7880 - val_accuracy: 0.7067
Epoch 114/300
218/218 [==============================] - 4s 17ms/step - loss: 0.6446 - ac
curacy: 0.7639 - val_loss: 0.7916 - val_accuracy: 0.7000
Epoch 115/300
218/218 [==============================] - 3s 15ms/step - loss: 0.6356 - ac
curacy: 0.7677 - val_loss: 0.8291 - val_accuracy: 0.7013
Epoch 116/300
218/218 [==============================] - 3s 16ms/step - loss: 0.6399 - ac
curacy: 0.7704 - val_loss: 0.8235 - val_accuracy: 0.6893
Epoch 117/300
218/218 [==============================] - 3s 16ms/step - loss: 0.6323 - ac
curacy: 0.7705 - val_loss: 0.7894 - val_accuracy: 0.7153
Epoch 118/300
218/218 [==============================] - 3s 16ms/step - loss: 0.6352 - ac
curacy: 0.7695 - val_loss: 0.7969 - val_accuracy: 0.6993
Epoch 119/300
218/218 [==============================] - 3s 16ms/step - loss: 0.6287 - ac
curacy: 0.7707 - val_loss: 0.8088 - val_accuracy: 0.6880
Epoch 120/300
218/218 [==============================] - 3s 15ms/step - loss: 0.6270 - ac
curacy: 0.7733 - val_loss: 0.8398 - val_accuracy: 0.6927
Epoch 121/300
218/218 [==============================] - 3s 16ms/step - loss: 0.6174 - ac
curacy: 0.7756 - val_loss: 0.7783 - val_accuracy: 0.7027
Epoch 122/300
```

```
218/218 [==============================] - 3s 15ms/step - loss: 0.6261 - ac
curacy: 0.7760 - val_loss: 0.7977 - val_accuracy: 0.7047
Epoch 123/300
218/218 [==============================] - 3s 16ms/step - loss: 0.6321 - ac
curacy: 0.7733 - val_loss: 0.7787 - val_accuracy: 0.7060
Epoch 124/300
218/218 [==============================] - 3s 15ms/step - loss: 0.6285 - ac
curacy: 0.7749 - val_loss: 0.8206 - val_accuracy: 0.6953
Epoch 125/300
218/218 [==============================] - 3s 15ms/step - loss: 0.6201 - ac
curacy: 0.7744 - val_loss: 0.7899 - val_accuracy: 0.6967
Epoch 126/300
218/218 [==============================] - 3s 16ms/step - loss: 0.6200 - ac
curacy: 0.7713 - val_loss: 0.8000 - val_accuracy: 0.6867
Epoch 127/300
218/218 [==============================] - 3s 15ms/step - loss: 0.6078 - ac
curacy: 0.7828 - val_loss: 0.7818 - val_accuracy: 0.6967
Epoch 128/300
218/218 [==============================] - 3s 15ms/step - loss: 0.6033 - ac
curacy: 0.7795 - val_loss: 0.7662 - val_accuracy: 0.7213
Epoch 129/300
218/218 [==============================] - 3s 15ms/step - loss: 0.6042 - ac
curacy: 0.7795 - val_loss: 0.8351 - val_accuracy: 0.6880
Epoch 130/300
218/218 [==============================] - 3s 15ms/step - loss: 0.6312 - ac
curacy: 0.7667 - val_loss: 0.8173 - val_accuracy: 0.6953
Epoch 131/300
218/218 [==============================] - 3s 15ms/step - loss: 0.6114 - ac
curacy: 0.7815 - val_loss: 0.7965 - val_accuracy: 0.7020
Epoch 132/300
218/218 [==============================] - 3s 15ms/step - loss: 0.6050 - ac
curacy: 0.7792 - val_loss: 0.7894 - val_accuracy: 0.6927
Epoch 133/300
218/218 [==============================] - 3s 13ms/step - loss: 0.6054 - ac
curacy: 0.7774 - val_loss: 0.8047 - val_accuracy: 0.6953
Epoch 134/300
218/218 [==============================] - 2s 11ms/step - loss: 0.5974 - ac
curacy: 0.7859 - val_loss: 0.8263 - val_accuracy: 0.7027
Epoch 135/300
218/218 [==============================] - 3s 12ms/step - loss: 0.6100 - ac
curacy: 0.7750 - val_loss: 0.7890 - val_accuracy: 0.7167
Epoch 136/300
218/218 [==============================] - 2s 11ms/step - loss: 0.5921 - ac
curacy: 0.7830 - val_loss: 0.7721 - val_accuracy: 0.7180
Epoch 137/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5916 - ac
curacy: 0.7879 - val_loss: 0.7850 - val_accuracy: 0.7107
Epoch 138/300
218/218 [==============================] - 2s 11ms/step - loss: 0.6084 - ac
curacy: 0.7787 - val_loss: 0.7939 - val_accuracy: 0.7053
Epoch 139/300
218/218 [==============================] - 2s 11ms/step - loss: 0.5866 - ac
curacy: 0.7944 - val_loss: 0.7638 - val_accuracy: 0.7127
Epoch 140/300
218/218 [==============================] - 2s 11ms/step - loss: 0.6029 - ac
curacy: 0.7779 - val_loss: 0.8294 - val_accuracy: 0.6880
Epoch 141/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5926 - ac
curacy: 0.7815 - val_loss: 0.8059 - val_accuracy: 0.7060
```

```
Epoch 142/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5916 - ac
curacy: 0.7874 - val_loss: 0.7802 - val_accuracy: 0.7040
Epoch 143/300
218/218 [==============================] - 2s 11ms/step - loss: 0.5994 - ac
curacy: 0.7813 - val_loss: 0.7834 - val_accuracy: 0.7080
Epoch 144/300
218/218 [==============================] - 2s 11ms/step - loss: 0.5721 - ac
curacy: 0.7895 - val_loss: 0.7932 - val_accuracy: 0.7047
Epoch 145/300
218/218 [==============================] - 3s 11ms/step - loss: 0.5865 - ac
curacy: 0.7895 - val_loss: 0.8439 - val_accuracy: 0.6853
Epoch 146/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5857 - ac
curacy: 0.7885 - val_loss: 0.8054 - val_accuracy: 0.7013
Epoch 147/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5807 - ac
curacy: 0.7902 - val_loss: 0.7850 - val_accuracy: 0.7087
Epoch 148/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5854 - ac
curacy: 0.7866 - val_loss: 0.7767 - val_accuracy: 0.7027
Epoch 149/300
218/218 [==============================] - 3s 13ms/step - loss: 0.5997 - ac
curacy: 0.7816 - val_loss: 0.7935 - val_accuracy: 0.7120
Epoch 150/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5708 - ac
curacy: 0.7886 - val_loss: 0.8585 - val_accuracy: 0.6927
Epoch 151/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5794 - ac
curacy: 0.7951 - val_loss: 0.7866 - val_accuracy: 0.7033
Epoch 152/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5829 - ac
curacy: 0.7898 - val_loss: 0.7841 - val_accuracy: 0.7140
Epoch 153/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5800 - ac
curacy: 0.7944 - val_loss: 0.7745 - val_accuracy: 0.6967
Epoch 154/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5783 - ac
curacy: 0.7889 - val_loss: 0.7824 - val_accuracy: 0.7047
Epoch 155/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5819 - ac
curacy: 0.7901 - val_loss: 0.8203 - val_accuracy: 0.6933
Epoch 156/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5665 - ac
curacy: 0.7915 - val_loss: 0.7891 - val_accuracy: 0.7087
Epoch 157/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5760 - ac
curacy: 0.7918 - val_loss: 0.7837 - val_accuracy: 0.6980
Epoch 158/300
218/218 [==============================] - 3s 13ms/step - loss: 0.5771 - ac
curacy: 0.7974 - val_loss: 0.7694 - val_accuracy: 0.7180
Epoch 159/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5785 - ac
curacy: 0.7868 - val_loss: 0.8073 - val_accuracy: 0.6887
Epoch 160/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5607 - ac
curacy: 0.7950 - val_loss: 0.8216 - val_accuracy: 0.7007
Epoch 161/300
218/218 [==============================] - 3s 15ms/step - loss: 0.5783 - ac
```

```
curacy: 0.7961 - val_loss: 0.8311 - val_accuracy: 0.6933
Epoch 162/300
218/218 [==============================] - 3s 13ms/step - loss: 0.5637 - ac
curacy: 0.7895 - val_loss: 0.7625 - val_accuracy: 0.7060
Epoch 163/300
218/218 [==============================] - 3s 16ms/step - loss: 0.5718 - ac
curacy: 0.7940 - val_loss: 0.8056 - val_accuracy: 0.6900
Epoch 164/300
218/218 [==============================] - 3s 15ms/step - loss: 0.5590 - ac
curacy: 0.7947 - val_loss: 0.7888 - val_accuracy: 0.6980
Epoch 165/300
218/218 [==============================] - 3s 16ms/step - loss: 0.5578 - ac
curacy: 0.7950 - val_loss: 0.7757 - val_accuracy: 0.6993
Epoch 166/300
218/218 [==============================] - 3s 16ms/step - loss: 0.5511 - ac
curacy: 0.7999 - val_loss: 0.8030 - val_accuracy: 0.6847
Epoch 167/300
218/218 [==============================] - 3s 16ms/step - loss: 0.5512 - ac
curacy: 0.8065 - val_loss: 0.8061 - val_accuracy: 0.6960
Epoch 168/300
218/218 [==============================] - 3s 16ms/step - loss: 0.5341 - ac
curacy: 0.8003 - val_loss: 0.7774 - val_accuracy: 0.7080
Epoch 169/300
218/218 [==============================] - 4s 17ms/step - loss: 0.5617 - ac
curacy: 0.7932 - val_loss: 0.7573 - val_accuracy: 0.7193
Epoch 170/300
218/218 [==============================] - 3s 15ms/step - loss: 0.5379 - ac
curacy: 0.8119 - val_loss: 0.7899 - val_accuracy: 0.7140
Epoch 171/300
218/218 [==============================] - 3s 16ms/step - loss: 0.5575 - ac
curacy: 0.8045 - val_loss: 0.7925 - val_accuracy: 0.7067
Epoch 172/300
218/218 [==============================] - 3s 15ms/step - loss: 0.5377 - ac
curacy: 0.8089 - val_loss: 0.8021 - val_accuracy: 0.7047
Epoch 173/300
218/218 [==============================] - 3s 16ms/step - loss: 0.5500 - ac
curacy: 0.8022 - val_loss: 0.7653 - val_accuracy: 0.7207
Epoch 174/300
218/218 [==============================] - 3s 15ms/step - loss: 0.5504 - ac
curacy: 0.8055 - val_loss: 0.8066 - val_accuracy: 0.7140
Epoch 175/300
218/218 [==============================] - 3s 15ms/step - loss: 0.5645 - ac
curacy: 0.7961 - val_loss: 0.8357 - val_accuracy: 0.6860
Epoch 176/300
218/218 [==============================] - 4s 16ms/step - loss: 0.5459 - ac
curacy: 0.8013 - val_loss: 0.7636 - val_accuracy: 0.7167
Epoch 177/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5536 - ac
curacy: 0.7983 - val_loss: 0.7931 - val_accuracy: 0.7167
Epoch 178/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5616 - ac
curacy: 0.7938 - val_loss: 0.7987 - val_accuracy: 0.6960
Epoch 179/300
218/218 [==============================] - 2s 11ms/step - loss: 0.5290 - ac
curacy: 0.8059 - val_loss: 0.8029 - val_accuracy: 0.7133
Epoch 180/300
218/218 [==============================] - 2s 11ms/step - loss: 0.5385 - ac
curacy: 0.8059 - val_loss: 0.7840 - val_accuracy: 0.7120
Epoch 181/300
```

```
218/218 [==============================] - 2s 11ms/step - loss: 0.5422 - ac
curacy: 0.8014 - val_loss: 0.7359 - val_accuracy: 0.7247
Epoch 182/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5549 - ac
curacy: 0.7993 - val_loss: 0.7671 - val_accuracy: 0.7200
Epoch 183/300
218/218 [==============================] - 2s 11ms/step - loss: 0.5260 - ac
curacy: 0.8132 - val_loss: 0.7879 - val_accuracy: 0.7180
Epoch 184/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5362 - ac
curacy: 0.8065 - val_loss: 0.7922 - val_accuracy: 0.7180
Epoch 185/300
218/218 [==============================] - 2s 11ms/step - loss: 0.5263 - ac
curacy: 0.8125 - val_loss: 0.7671 - val_accuracy: 0.7233
Epoch 186/300
218/218 [==============================] - 2s 11ms/step - loss: 0.5230 - ac
curacy: 0.8101 - val_loss: 0.7806 - val_accuracy: 0.7240
Epoch 187/300
218/218 [==============================] - 2s 11ms/step - loss: 0.5272 - ac
curacy: 0.8109 - val_loss: 0.8070 - val_accuracy: 0.7167
Epoch 188/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5262 - ac
curacy: 0.8115 - val_loss: 0.7815 - val_accuracy: 0.7080
Epoch 189/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5338 - ac
curacy: 0.8036 - val_loss: 0.7698 - val_accuracy: 0.7213
Epoch 190/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5446 - ac
curacy: 0.8066 - val_loss: 0.7576 - val_accuracy: 0.7187
Epoch 191/300
218/218 [==============================] - 2s 11ms/step - loss: 0.5311 - ac
curacy: 0.8096 - val_loss: 0.7833 - val_accuracy: 0.7113
Epoch 192/300
218/218 [==============================] - 2s 11ms/step - loss: 0.5362 - ac
curacy: 0.8098 - val_loss: 0.7433 - val_accuracy: 0.7220
Epoch 193/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5265 - ac
curacy: 0.8106 - val_loss: 0.7677 - val_accuracy: 0.7053
Epoch 194/300
218/218 [==============================] - 2s 11ms/step - loss: 0.5347 - ac
curacy: 0.8075 - val_loss: 0.7814 - val_accuracy: 0.7180
Epoch 195/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5383 - ac
curacy: 0.8069 - val_loss: 0.7632 - val_accuracy: 0.7200
Epoch 196/300
218/218 [==============================] - 2s 11ms/step - loss: 0.5311 - ac
curacy: 0.8066 - val_loss: 0.7473 - val_accuracy: 0.7207
Epoch 197/300
218/218 [==============================] - 2s 11ms/step - loss: 0.5278 - ac
curacy: 0.8125 - val_loss: 0.8008 - val_accuracy: 0.7013
Epoch 198/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5082 - ac
curacy: 0.8116 - val_loss: 0.7281 - val_accuracy: 0.7227
Epoch 199/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5356 - ac
curacy: 0.8057 - val_loss: 0.7429 - val_accuracy: 0.7120
Epoch 200/300
218/218 [==============================] - 2s 11ms/step - loss: 0.5289 - ac
curacy: 0.8057 - val_loss: 0.7806 - val_accuracy: 0.7113
```

```
Epoch 201/300
218/218 [==============================] - 2s 11ms/step - loss: 0.5092 - ac
curacy: 0.8164 - val_loss: 0.7941 - val_accuracy: 0.7020
Epoch 202/300
218/218 [==============================] - 2s 11ms/step - loss: 0.5143 - ac
curacy: 0.8159 - val_loss: 0.7493 - val_accuracy: 0.7140
Epoch 203/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5207 - ac
curacy: 0.8072 - val_loss: 0.7328 - val_accuracy: 0.7273
Epoch 204/300
218/218 [==============================] - 2s 11ms/step - loss: 0.5295 - ac
curacy: 0.8063 - val_loss: 0.7353 - val_accuracy: 0.7273
Epoch 205/300
218/218 [==============================] - 2s 11ms/step - loss: 0.5053 - ac
curacy: 0.8190 - val_loss: 0.7600 - val_accuracy: 0.7113
Epoch 206/300
218/218 [==============================] - 2s 11ms/step - loss: 0.5125 - ac
curacy: 0.8164 - val_loss: 0.8007 - val_accuracy: 0.7040
Epoch 207/300
218/218 [==============================] - 3s 11ms/step - loss: 0.5160 - ac
curacy: 0.8125 - val_loss: 0.7882 - val_accuracy: 0.7033
Epoch 208/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5303 - ac
curacy: 0.8063 - val_loss: 0.7639 - val_accuracy: 0.7100
Epoch 209/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5073 - ac
curacy: 0.8210 - val_loss: 0.7992 - val_accuracy: 0.7080
Epoch 210/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5115 - ac
curacy: 0.8193 - val_loss: 0.7941 - val_accuracy: 0.7027
Epoch 211/300
218/218 [==============================] - 3s 12ms/step - loss: 0.4991 - ac
curacy: 0.8207 - val_loss: 0.7627 - val_accuracy: 0.7167
Epoch 212/300
218/218 [==============================] - 2s 11ms/step - loss: 0.5105 - ac
curacy: 0.8187 - val_loss: 0.7915 - val_accuracy: 0.7020
Epoch 213/300
218/218 [==============================] - 2s 11ms/step - loss: 0.4960 - ac
curacy: 0.8175 - val_loss: 0.7536 - val_accuracy: 0.7200
Epoch 214/300
218/218 [==============================] - 2s 11ms/step - loss: 0.4911 - ac
curacy: 0.8254 - val_loss: 0.7320 - val_accuracy: 0.7260
Epoch 215/300
218/218 [==============================] - 2s 11ms/step - loss: 0.5054 - ac
curacy: 0.8171 - val_loss: 0.7580 - val_accuracy: 0.7240
Epoch 216/300
218/218 [==============================] - 2s 11ms/step - loss: 0.5013 - ac
curacy: 0.8190 - val_loss: 0.7749 - val_accuracy: 0.7067
Epoch 217/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5023 - ac
curacy: 0.8205 - val_loss: 0.7923 - val_accuracy: 0.7187
Epoch 218/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5119 - ac
curacy: 0.8157 - val_loss: 0.7317 - val_accuracy: 0.7247
Epoch 219/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5017 - ac
curacy: 0.8198 - val_loss: 0.7757 - val_accuracy: 0.7053
Epoch 220/300
218/218 [==============================] - 2s 11ms/step - loss: 0.4942 - ac
```

```
curacy: 0.8234 - val_loss: 0.7727 - val_accuracy: 0.7187
Epoch 221/300
218/218 [==============================] - 2s 11ms/step - loss: 0.4915 - ac
curacy: 0.8273 - val_loss: 0.7725 - val_accuracy: 0.7160
Epoch 222/300
218/218 [==============================] - 3s 12ms/step - loss: 0.5070 - ac
curacy: 0.8191 - val_loss: 0.7453 - val_accuracy: 0.7120
Epoch 223/300
214/218 [============================>.] - ETA: 0s - loss: 0.4946 - accurac
y: 0.8245
Reached 73.00% accuracy, so stopping training!!
218/218 [==============================] - 3s 12ms/step - loss: 0.4953 - ac
curacy: 0.8244 - val_loss: 0.7651 - val_accuracy: 0.7300
```

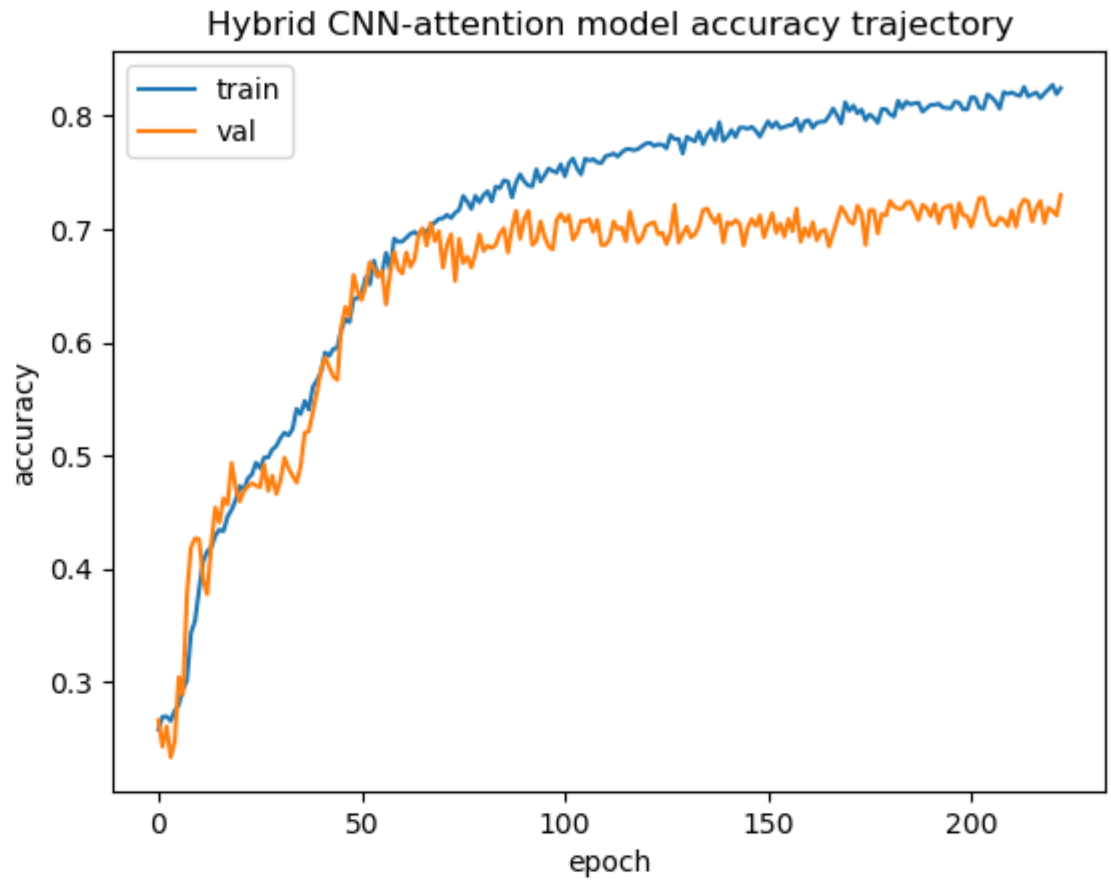
(vi) (CNN-MHA) Visualizing the accuracy and loss trajectory

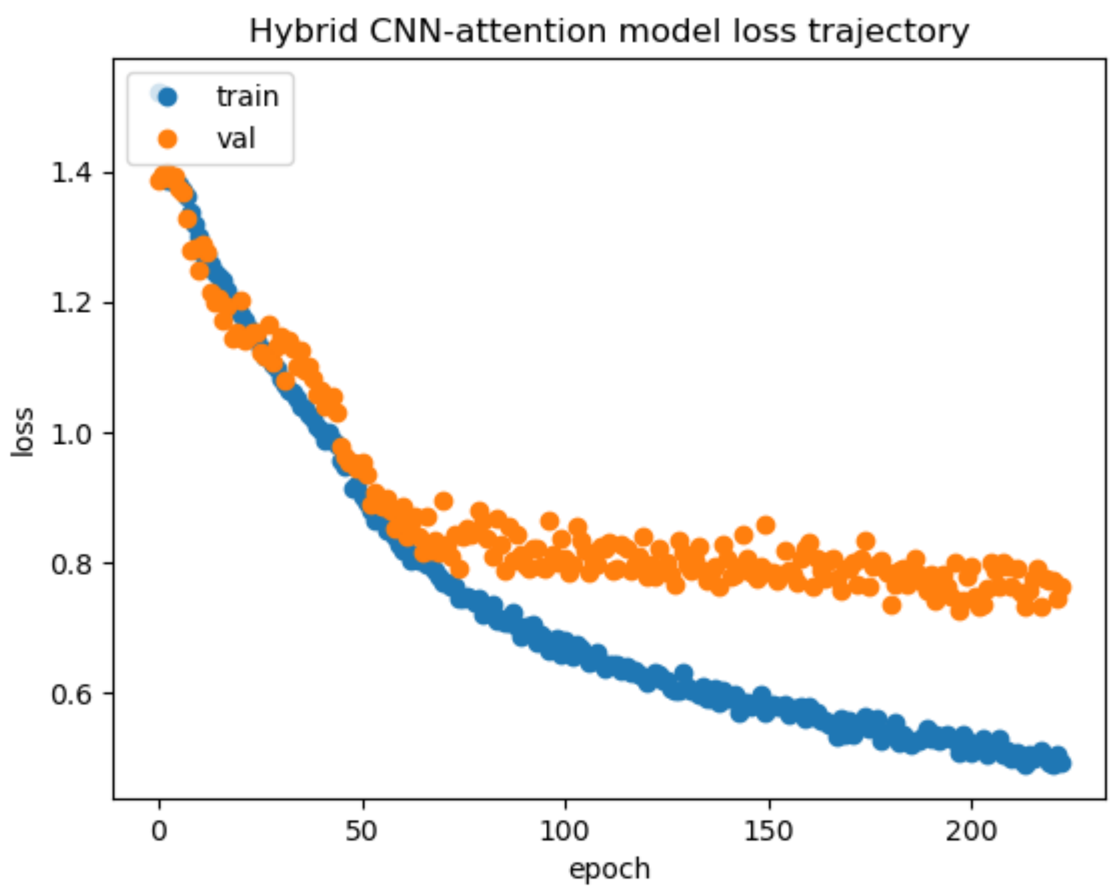
In [ ]:

```
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-attention 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-attention model loss trajectory')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```


Hybrid CNN-attention model accuracy trajectory
train
val
accuracy
0.8
0.7
0.6
0.5
0.4
0.3
0
50
100
150
200
epoch


Hybrid CNN-attention model loss trajectory
train
val
loss
1.4
1.2
1.0
0.8
0.6
0
50
100
150
200
epoch

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

In [ ]:
```
## Testing the hybrid CNN-MHA model

hybrid_cnn_mha_score = hybrid_cnn_mha_model.evaluate(x_test, y_test, verbose
print('Test accuracy of the hybrid CNN-MHA model:',hybrid_cnn_mha_score[1])
```

```
Test accuracy of the hybrid CNN-attention model: 0.707110583782196
```

In [ ]: