**Model Training**

[38]

**import** tensorflow **as** tf  
**from** tensorflow.keras.models **import** Sequential  
**from** tensorflow.keras.layers **import** Dense, Flatten, Dropout, LSTM  
  
**from** tensorflow.keras.callbacks **import** ModelCheckpoint, EarlyStopping

[39]

model = Sequential([  
    Flatten(input\_shape=(X\_train.shape[1], X\_train.shape[2])),  
    Dense(64, activation=**'relu'**),  
    Dense(128, activation=**'relu'**),  
    Dense(512, activation=**'relu'**),  
    Dense(512, activation=**'relu'**),  
    Dropout(rate=0.2),  
    Dense(64, activation=**'relu'**),  
    Dense(32, activation=**'relu'**),  
    Dense(y\_train.shape[1], activation=**'softmax'**),  
])  
  
model.summary()

Model: "sequential\_1"  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Layer (type) Output Shape Param #   
=================================================================  
flatten\_1 (Flatten) (None, 84) 0   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
dense\_7 (Dense) (None, 64) 5440   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
dense\_8 (Dense) (None, 128) 8320   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
dense\_9 (Dense) (None, 512) 66048   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
dense\_10 (Dense) (None, 512) 262656   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
dropout\_1 (Dropout) (None, 512) 0   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
dense\_11 (Dense) (None, 64) 32832   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
dense\_12 (Dense) (None, 32) 2080   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
dense\_13 (Dense) (None, 15) 495   
=================================================================  
Total params: 377,871  
Trainable params: 377,871  
Non-trainable params: 0  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

[40]

callback\_checkpoint = ModelCheckpoint(  
    **'./checkpoints/'**, monitor=**'val\_loss'**, verbose=0, save\_best\_only=**True**,  
    save\_weights\_only=**True**, mode=**'auto'**, save\_freq=**'epoch'**,  
)  
callback\_loss = EarlyStopping(monitor=**'loss'**, patience=4)  
  
model.compile(loss=**'categorical\_crossentropy'**, optimizer=**'adam'**, metrics=[**'accuracy'**])

[41]

history = model.fit(X\_train, y\_train,   
                    epochs=50,   
                    callbacks=[callback\_checkpoint, callback\_loss])

Epoch 1/50  
240/240 [==============================] - 2s 5ms/step - loss: 1.5844 - accuracy: 0.3947  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 2/50  
240/240 [==============================] - 1s 3ms/step - loss: 0.9934 - accuracy: 0.6137  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 3/50  
240/240 [==============================] - 1s 4ms/step - loss: 0.7049 - accuracy: 0.7324  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 4/50  
240/240 [==============================] - 1s 4ms/step - loss: 0.5005 - accuracy: 0.8185  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 5/50  
240/240 [==============================] - 1s 4ms/step - loss: 0.4246 - accuracy: 0.8450  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 6/50  
240/240 [==============================] - 1s 4ms/step - loss: 0.3675 - accuracy: 0.8722  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 7/50  
240/240 [==============================] - 1s 3ms/step - loss: 0.3012 - accuracy: 0.8993  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 8/50  
240/240 [==============================] - 1s 4ms/step - loss: 0.2763 - accuracy: 0.9050  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 9/50  
240/240 [==============================] - 1s 4ms/step - loss: 0.2369 - accuracy: 0.9271  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 10/50  
240/240 [==============================] - 1s 4ms/step - loss: 0.1889 - accuracy: 0.9445  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 11/50  
240/240 [==============================] - 1s 4ms/step - loss: 0.1823 - accuracy: 0.9467  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 12/50  
240/240 [==============================] - 1s 3ms/step - loss: 0.1582 - accuracy: 0.9554  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 13/50  
240/240 [==============================] - 1s 3ms/step - loss: 0.2254 - accuracy: 0.9390  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 14/50  
240/240 [==============================] - 1s 3ms/step - loss: 0.1409 - accuracy: 0.9625  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 15/50  
240/240 [==============================] - 1s 3ms/step - loss: 0.1182 - accuracy: 0.9668  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 16/50  
240/240 [==============================] - 1s 3ms/step - loss: 0.0979 - accuracy: 0.9744  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 17/50  
240/240 [==============================] - 1s 3ms/step - loss: 0.1480 - accuracy: 0.9600  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 18/50  
240/240 [==============================] - 1s 3ms/step - loss: 0.1918 - accuracy: 0.9420  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 19/50  
240/240 [==============================] - 1s 3ms/step - loss: 0.0952 - accuracy: 0.9745  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 20/50  
240/240 [==============================] - 1s 3ms/step - loss: 0.1408 - accuracy: 0.9635  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 21/50  
240/240 [==============================] - 1s 3ms/step - loss: 0.1487 - accuracy: 0.9562  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 22/50  
240/240 [==============================] - 1s 3ms/step - loss: 0.1015 - accuracy: 0.9701  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 23/50  
240/240 [==============================] - 1s 3ms/step - loss: 0.0891 - accuracy: 0.9758  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 24/50  
240/240 [==============================] - 1s 3ms/step - loss: 0.0764 - accuracy: 0.9811  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 25/50  
240/240 [==============================] - 1s 3ms/step - loss: 0.0860 - accuracy: 0.9786  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 26/50  
240/240 [==============================] - 1s 3ms/step - loss: 0.1304 - accuracy: 0.9650  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 27/50  
240/240 [==============================] - 1s 3ms/step - loss: 0.0975 - accuracy: 0.9752  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 28/50  
240/240 [==============================] - 1s 3ms/step - loss: 0.0695 - accuracy: 0.9835  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 29/50  
240/240 [==============================] - 1s 3ms/step - loss: 0.1027 - accuracy: 0.9722  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 30/50  
240/240 [==============================] - 1s 3ms/step - loss: 0.0924 - accuracy: 0.9756  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 31/50  
240/240 [==============================] - 1s 3ms/step - loss: 0.0968 - accuracy: 0.9753  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.  
Epoch 32/50  
240/240 [==============================] - 1s 3ms/step - loss: 0.0806 - accuracy: 0.9766  
WARNING:tensorflow:Can save best model only with val\_loss available, skipping.

[46]

*# model.save("./checkpoints/alphabets\_angles\_model\_2")*  
model.save(**"./checkpoints/gesture\_15\_model\_v2"**)

INFO:tensorflow:Assets written to: ./checkpoints/gesture\_15\_model\_v2\assets