**Model Evaluation and Metrics**

[42]

**from** sklearn.metrics **import** classification\_report, confusion\_matrix  
**import** matplotlib.pyplot **as** plt  
**import** seaborn **as** sns

[140]

**from** tensorflow.keras.models **import** load\_model  
model = load\_model(**"checkpoints/two\_hand\_9\_model"**)  
*# model = load\_model("checkpoints/alphabets\_Z\_model")*

[43]

y\_pred = model.predict(X\_test)  
y\_preds = np.argmax(y\_pred, axis=1)  
  
unique, counts = np.unique(y\_preds, return\_counts=**True**)  
print(dict(zip(unique, counts)))

{0: 158, 1: 150, 2: 149, 3: 149, 4: 146, 5: 148, 6: 159, 7: 148, 8: 128, 9: 156, 10: 155, 11: 21, 12: 79, 13: 21, 14: 147}

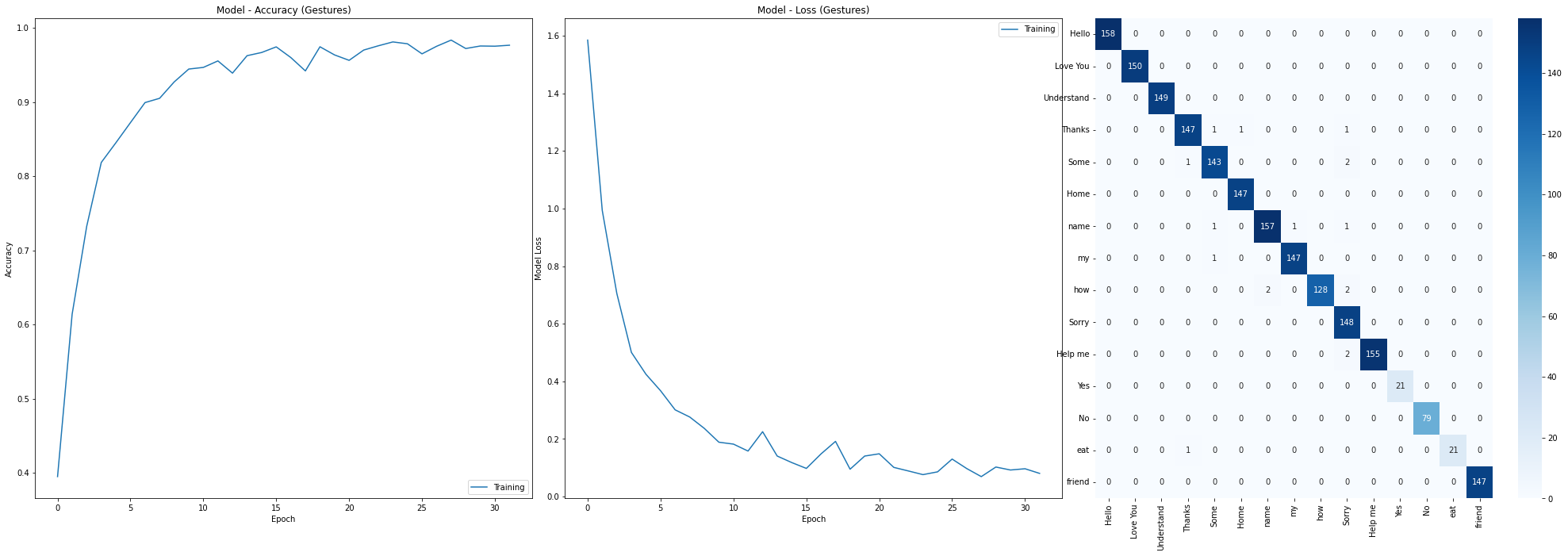
[44]

**def** evaluate\_model(history, X\_test, y\_test, model, title):  
  **"""**  
**Input:**  
**history - Model history variable**  
**X\_test - Test features**  
**y\_test - Target test variables**  
**model - Keras Model**  
  
**Output:**  
**Accuracy Plot (Training + Validation)**  
**Loss Plot (Training + Validation)**  
**Confusion matrix**  
**Accuracy**  
**Classification Report**  
**"""**    
  scores = model.evaluate((X\_test),y\_test, verbose=0)  
    
  fig, axs = plt.subplots(1,3, figsize=(28,9))  
  fig.tight\_layout()  
  
  axs[0].plot(history.history[**'accuracy'**])  
  axs[0].set\_xlabel(**'Epoch'**)  
  axs[0].set\_ylabel(**'Accuracy'**)  
  axs[0].set\_title(**'Model - Accuracy ('** + title +**")"**)  
  axs[0].legend([**'Training'**], loc=**'lower right'**)  
  
  axs[1].plot(history.history[**'loss'**])  
  axs[1].set\_xlabel(**'Epoch'**)  
  axs[1].set\_ylabel(**'Model Loss'**)  
  axs[1].set\_title(**'Model - Loss ('** + title +**")"**)  
  axs[1].legend([**'Training'**], loc=**'upper right'**)  
  
  target\_names=actions  
    
  y\_true=[]  
  **for** element **in** y\_test:  
      y\_true.append(np.argmax(element))  
  prediction\_proba = model.predict(X\_test)  
  prediction = np.argmax(prediction\_proba,axis=1)  
  cnf\_matrix = confusion\_matrix(y\_true, prediction)      
  g = sns.heatmap(cnf\_matrix, annot=**True**, cmap=**'Blues'**, fmt=**'g'**, ax=axs[2],  
                  xticklabels=target\_names, yticklabels=target\_names)  
  plt.show()  
  
  print(**"\nAccuracy: %.2f%% \n"** % (scores[1]\*100))  
  print(classification\_report(y\_true, prediction, target\_names=target\_names))

[45]

evaluate\_model(history, X\_test, y\_test, model, title=**"Gestures"**)

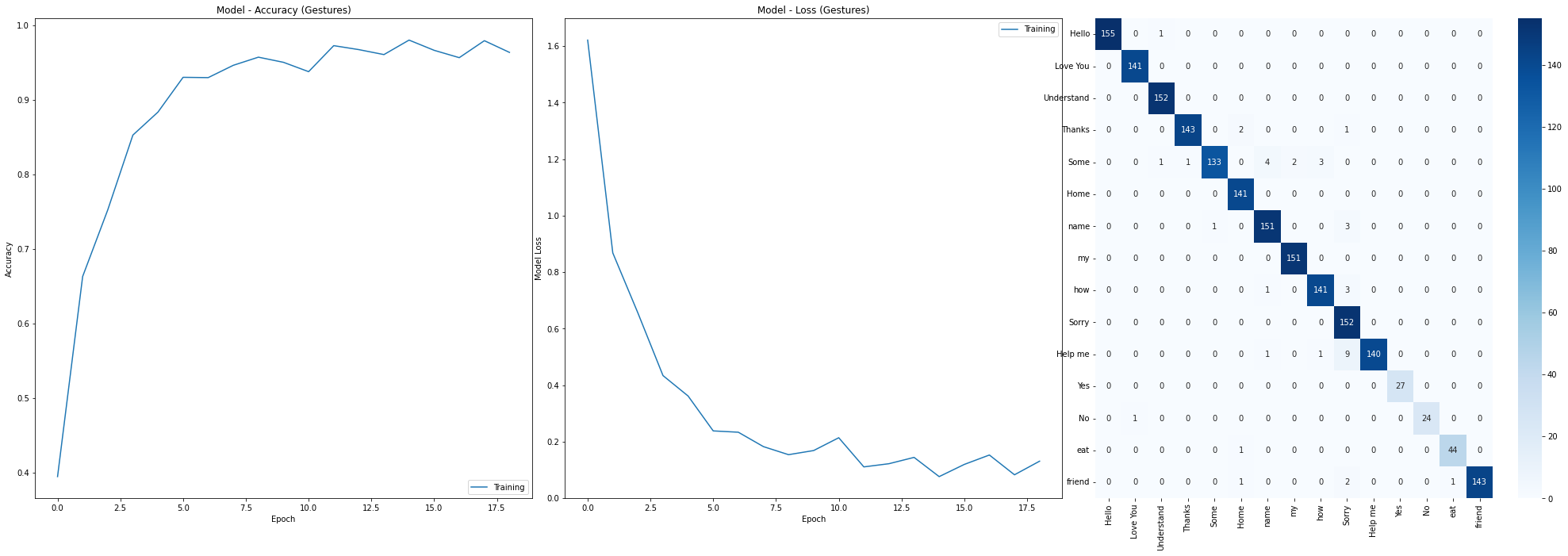
Accuracy: 99.11%   
  
 precision recall f1-score support  
  
 Hello 1.00 1.00 1.00 158  
 Love You 1.00 1.00 1.00 150  
 Understand 1.00 1.00 1.00 149  
 Thanks 0.99 0.98 0.98 150  
 Some 0.98 0.98 0.98 146  
 Home 0.99 1.00 1.00 147  
 name 0.99 0.98 0.98 160  
 my 0.99 0.99 0.99 148  
 how 1.00 0.97 0.98 132  
 Sorry 0.95 1.00 0.97 148  
 Help me 1.00 0.99 0.99 157  
 Yes 1.00 1.00 1.00 21  
 No 1.00 1.00 1.00 79  
 eat 1.00 0.95 0.98 22  
 friend 1.00 1.00 1.00 147  
  
 accuracy 0.99 1914  
 macro avg 0.99 0.99 0.99 1914  
weighted avg 0.99 0.99 0.99 1914



[21]

evaluate\_model(history, X\_test, y\_test, model, title=**"Gestures"**)

Accuracy: 97.87%   
  
 precision recall f1-score support  
  
 Hello 1.00 0.99 1.00 156  
 Love You 0.99 1.00 1.00 141  
 Understand 0.99 1.00 0.99 152  
 Thanks 0.99 0.98 0.99 146  
 Some 0.99 0.92 0.96 144  
 Home 0.97 1.00 0.99 141  
 name 0.96 0.97 0.97 155  
 my 0.99 1.00 0.99 151  
 how 0.97 0.97 0.97 145  
 Sorry 0.89 1.00 0.94 152  
 Help me 1.00 0.93 0.96 151  
 Yes 1.00 1.00 1.00 27  
 No 1.00 0.96 0.98 25  
 eat 0.98 0.98 0.98 45  
 friend 1.00 0.97 0.99 147  
  
 accuracy 0.98 1878  
 macro avg 0.98 0.98 0.98 1878  
weighted avg 0.98 0.98 0.98 1878



[242]

evaluate\_model(history, X\_test, y\_test, model, title=**"Alphabets"**)

Accuracy: 98.07%   
  
 precision recall f1-score support  
  
 A 0.97 0.99 0.98 149  
 B 0.99 0.99 0.99 144  
 C 0.98 1.00 0.99 147  
 D 0.86 0.99 0.92 147  
 E 0.99 0.84 0.91 150  
 F 1.00 0.99 1.00 146  
 G 1.00 1.00 1.00 149  
 H 1.00 1.00 1.00 151  
 I 0.94 1.00 0.97 158  
 J 0.98 0.93 0.95 136  
 K 1.00 0.98 0.99 154  
 L 1.00 1.00 1.00 137  
 M 0.99 0.99 0.99 162  
 N 0.99 0.98 0.99 161  
 O 1.00 0.99 0.99 150  
 P 0.98 1.00 0.99 128  
 Q 1.00 0.99 1.00 147  
 R 1.00 0.96 0.98 155  
 U 0.91 1.00 0.96 129  
 V 0.99 0.96 0.98 167  
 W 0.99 0.97 0.98 145  
 X 1.00 1.00 1.00 148  
 Y 1.00 1.00 1.00 155  
  
 accuracy 0.98 3415  
 macro avg 0.98 0.98 0.98 3415  
weighted avg 0.98 0.98 0.98 3415

