Signal Modulation Classification

Nourhan Waleed 6609
Mohamed Zaytoon 6670
Mostafa Twfiq 6672

June 25, 2022

1 Importing packages

[2]:
```
import pickle
import os
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from scipy import integrate
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import normalize
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Activation, Flatten, Conv1D,␣
↪MaxPool1D, Conv2D, MaxPool2D,BatchNormalization, GRU,␣
↪TimeDistributed,ConvLSTM2D
from tensorflow.keras.optimizers import Adam
from sklearn import metrics
import tensorflow as tf
from sklearn.preprocessing import OneHotEncoder
from tensorflow.keras.callbacks import ModelCheckpoint
from sklearn.preprocessing import OneHotEncoder
from keras.layers import LSTM
from keras.layers.embeddings import Embedding
from keras.layers import SimpleRNN
from sklearn.metrics import accuracy_score
```

2 1. Download the Dataset

[3]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
```

```
drive.mount("/content/drive", force_remount=True).
```

[4]:
```
import os
os.chdir('/content/drive/MyDrive/RML2016_dataset/')
Data = pickle.load(open("RML2016.10b.dat",'rb'), encoding = 'bytes')
snrs, mods = map(lambda j: sorted(list(set(map(lambda x: x[j], Data.keys())))),␣
 ↪[1,0])
```

[4]:
```
print(f"Mods: {mods}")
print(f"SNRs: {snrs}")
```

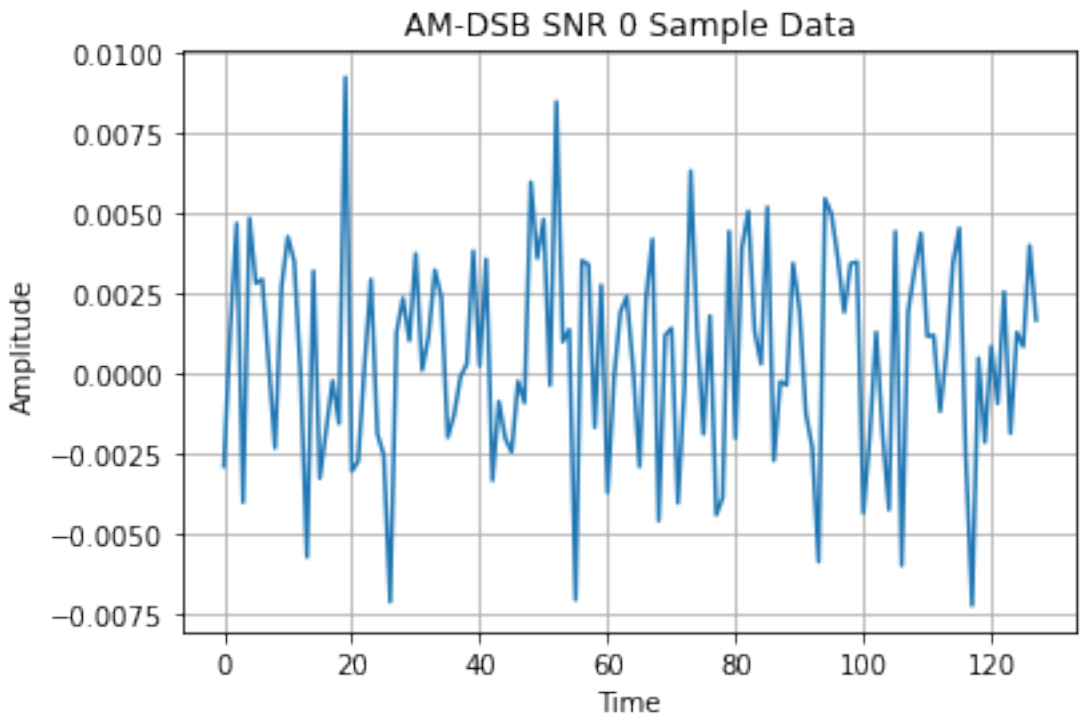
```
Mods: [b'8PSK', b'AM-DSB', b'BPSK', b'CPFSK', b'GFSK', b'PAM4', b'QAM16',
b'QAM64', b'QPSK', b'WBFM']
SNRs: [-20, -18, -16, -14, -12, -10, -8, -6, -4, -2, 0, 2, 4, 6, 8, 10, 12, 14,
16, 18]
```

[5]:
```
samples = []
snr_mod_labels = []
for snr in snrs:
  x = []
  y = []
  curr_snr_labels = []
  for mod in mods:
    x.append(Data[(mod,snr)])
    y.append([mod] * Data[(mod,snr)].shape[0])
    curr_snr_labels.append([snr, mod] * Data[(mod,snr)].shape[0])
  y = np.array(y)
  curr_snr_labels = np.array(curr_snr_labels)
  snr_mod_labels.append(curr_snr_labels.reshape((int(curr_snr_labels.shape[1]/2)␣
 ↪* curr_snr_labels.shape[0], 2)))
  samples.append(np.vstack(x))

samples = np.array(samples)
snr_mod_labels = np.array(snr_mod_labels)
```

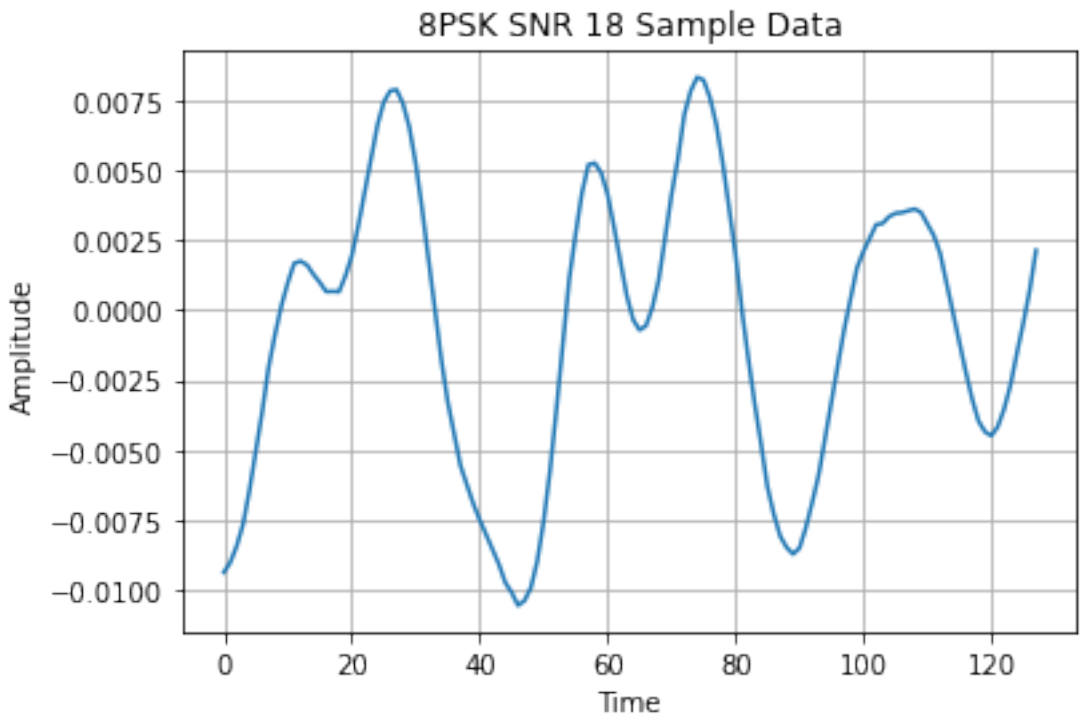
[6]:
```
print(samples.shape)
print(snr_mod_labels.shape)
```

```
(20, 60000, 2, 128)
(20, 60000, 2)
```

[ ]:
```
plt.plot(Data[b'AM-DSB',0][2,0])
plt.xlabel("Time")
plt.ylabel("Amplitude")
plt.title("AM-DSB SNR 0 Sample Data")
plt.grid(b=True, axis='both')
```



[ ]:
```
plt.plot(Data[b'8PSK',18][2,0])
plt.xlabel("Time")
plt.ylabel("Amplitude")
plt.title("8PSK SNR 18 Sample Data")
plt.grid(b=True, axis='both')
```



3 Data Splitting And Balancing

[6]:
```
#Reshaping dataset
samples = samples.reshape((samples.shape[0] * samples.shape[1], samples.
 ↪shape[2], samples.shape[3]))
labels = snr_mod_labels.reshape((-1, 2)) # Notic that the labels are at this␣
 ↪format [snr, modulation]
```

[ ]:
```
print('samples shape:', samples.shape)
print('labels shape:', labels.shape)
```

```
samples shape: (1200000, 2, 128)
labels shape: (1200000, 2)
```

[7]:
```
#Splitting data
training_val_data, testing_data, training_val_pair_labels, testing_pair_labels =␣
 ↪train_test_split(samples, labels,stratify=labels, shuffle=True, test_size=0.3)
training_data, validation_data, training_pair_labels, validation_pair_labels = ␣
 ↪train_test_split(training_val_data,␣
 ↪training_val_pair_labels,stratify=training_val_pair_labels, shuffle=True,␣
 ↪test_size=0.05)
```

[9]:
```
del samples
```

[8]:
```
training_labels = training_pair_labels[:, 1]
validation_labels = validation_pair_labels[:, 1]
```

[ ]:
```
print('training data shape:', training_data.shape)
print('training labels shape:', training_labels.shape)
print('validation data shape:', validation_data.shape)
print('validation labels shape:', validation_labels.shape)
print('testing data shape:', testing_data.shape)
print('testing labels shape:', testing_pair_labels.shape)
```

```
training data shape: (798000, 2, 128)
training labels shape: (798000,)
validation data shape: (42000, 2, 128)
validation labels shape: (42000,)
testing data shape: (360000, 2, 128)
testing labels shape: (360000, 2)
```

[49]:
```
unique, counts = np.unique(training_labels, return_counts=True)
print('training:\t', dict(zip(unique, counts)))
unique, counts = np.unique(validation_labels, return_counts=True)
print('validation:\t', dict(zip(unique, counts)))
unique, counts = np.unique(testing_pair_labels, return_counts=True)
print('testing:\t', dict(zip(unique, counts)))
```

```
training:        {b'8PSK': 79800, b'AM-DSB': 79800, b'BPSK': 79800, b'CPFSK':
79800, b'GFSK': 79800, b'PAM4': 79800, b'QAM16': 79800, b'QAM64': 79800,
b'QPSK': 79800, b'WBFM': 79800}
validation:      {b'8PSK': 4200, b'AM-DSB': 4200, b'BPSK': 4200, b'CPFSK': 4200,
b'GFSK': 4200, b'PAM4': 4200, b'QAM16': 4200, b'QAM64': 4200, b'QPSK': 4200,
b'WBFM': 4200}
testing:         {b'-10': 18000, b'-12': 18000, b'-14': 18000, b'-16': 18000,
b'-18': 18000, b'-2': 18000, b'-20': 18000, b'-4': 18000, b'-6': 18000, b'-8':
18000, b'0': 18000, b'10': 18000, b'12': 18000, b'14': 18000, b'16': 18000,
b'18': 18000, b'2': 18000, b'4': 18000, b'6': 18000, b'8': 18000, b'8PSK':
36000, b'AM-DSB': 36000, b'BPSK': 36000, b'CPFSK': 36000, b'GFSK': 36000,
b'PAM4': 36000, b'QAM16': 36000, b'QAM64': 36000, b'QPSK': 36000, b'WBFM':
36000}
```

[10]:
```
# converting labels to one hot encoding
training_onehot = OneHotEncoder(sparse = False).fit_transform(training_labels.
 ↪reshape(-1,1))
validation_onehot = OneHotEncoder(sparse = False).
 ↪fit_transform(validation_labels.reshape(-1,1))
```

[ ]:
```
print('training onehot encoding shape:', training_onehot.shape)
print('validation onehot encoding shape:', validation_onehot.shape)
```

```
training onehot encoding shape: (798000, 10)
validation onehot encoding shape: (42000, 10)
```

4 Scoring functions

[11]:
```
def model_pred_and_accuracy(model, testing_data, testing_labels):
  pred = model.predict(testing_data)
  decoded_pred = np.argmax(pred, axis=1)
  testing_labels = np.argmax(testing_labels, axis=1)

  return pred, accuracy_score(testing_labels, decoded_pred)
```

[12]:
```
def confusion_matrixf(pred, actual, title):
  actual_decode = np.argmax(actual, axis=1)
  pred_decode = np.argmax(pred, axis=1)
  confusion_mtx = confusion_matrix(actual_decode, pred_decode)
  plt.figure(figsize=(10, 8))
  sns.heatmap(confusion_mtx, xticklabels=mods, yticklabels=mods, annot=True,␣
 ↪fmt='g')
  plt.xlabel('Prediction')
  plt.ylabel('Actual')
  plt.title(title)
  plt.show()
```

[13]:
```
def plot_model_history(history, title=''):
  pd.DataFrame(history.history).plot(figsize=(8,5))
  plt.title(title)
  plt.show()
```

[14]:
```
def plot_snr_vs_acc(snr, acc):
  plt.plot(snr, acc, label = "SNR vs Accuracy")
  plt.xlabel('SNR')
  plt.ylabel('Accuracy')
  plt.legend()
  plt.show()
```

[51]:
```
import gc, torch
def clear_cache():
  for _ in range(0,100):
    gc.collect()
    torch.cuda.empty_cache()
```

[50]:
```
clear_cache()
```

[52]:
```
def model_scoring(model, testing_data, testing_pair_labels):
  sorted_idx = testing_pair_labels[:, 0].astype(int).argsort() # sort by snr␣
↪value
  testing_data = testing_data[sorted_idx] # to sort samples by snr
  testing_pair_labels = testing_pair_labels[sorted_idx] # to sort labels by snr

  SNRs = testing_pair_labels[:, 0]
  modulations = testing_pair_labels[:, 1]
  unique_snr, snr_count = np.unique(SNRs, return_counts=True)
  snr_count_dict = dict(zip(unique, counts))

  curr_sample = 0
  snrArr = []
  acc = []
  for snr in range(len(unique_snr)):
    curr_snr = SNRs[curr_sample]
    curr_snr_samples = []
    curr_snr_labels = []
    for j in range(snr_count_dict[curr_snr]):
      curr_snr_samples.append(testing_data[curr_sample])
      curr_snr_labels.append(modulations[curr_sample])
      curr_sample += 1

    curr_snr_samples = np.array(curr_snr_samples)
    curr_snr_labels = np.array(curr_snr_labels)
    onehot_labels = OneHotEncoder(sparse = False).fit_transform(curr_snr_labels.
↪reshape(-1,1))

    pred, accuracy = model_pred_and_accuracy(model, curr_snr_samples,␣
↪onehot_labels)
    snrArr.append(curr_snr.decode())
    acc.append(accuracy)
    print('Accuracy at SNR = ' + curr_snr.decode() + ' is ' + str(accuracy) +␣
↪'%')
    confusion_matrixf(pred, onehot_labels, 'SNR: ' + curr_snr.decode())
    clear_cache()
  plot_snr_vs_acc(snrArr, acc)
```

5 Normal Feature Space

5.1 CNN LSTM Model

[16]:
```
learning_rate = 0.001
batch_size = 512
epochs = 200
```

```
[ ]: '''cnn_lstm_model = Sequential()
cnn_lstm_model.add(ConvLSTM2D(256, kernel_size = (3, 3), padding = 'same',␣
 ↪activation='relu', return_sequences = True))
cnn_lstm_model.add(Dropout(0.5))
cnn_lstm_model.add(ConvLSTM2D(64, kernel_size = (3, 3), padding = 'same',␣
 ↪activation='relu', return_sequences = True))
cnn_lstm_model.add(Dropout(0.5))
cnn_lstm_model.add(Flatten())
cnn_lstm_model.add(Dense(128, activation='relu' ))
cnn_lstm_model.add(Dense(10,  activation='softmax'))
cnn_lstm_model.compile(loss=tf.keras.losses.CategoricalCrossentropy(),␣
 ↪metrics=['accuracy'], optimizer=tf.keras.optimizers.
 ↪Adam(learning_rate=learning_rate))'''
```

```
[ ]: X_trainp = np.asarray(np.transpose(training_data, axes=(0,2,1)))
X_valp   = np.asarray(np.transpose(validation_data  , axes=(0,2,1)))
n_timesteps, n_features, n_outputs = X_trainp.shape[1], X_trainp.shape[2],␣
 ↪validation_onehot.shape[1]
n_steps, n_length = 4, 32
X_trainp = X_trainp.reshape((X_trainp.shape[0], n_steps, n_length, n_features))
X_valp = X_valp.reshape((X_valp.shape[0], n_steps, n_length, n_features))
```

```
[ ]: X_test = np.asarray(np.transpose(testing_data, axes=(0,2,1)))
n_timesteps, n_features, n_outputs = X_test.shape[1], X_test.shape[2],␣
 ↪validation_onehot.shape[1]
n_steps, n_length = 4, 32
X_test = X_test.reshape((X_test.shape[0], n_steps, n_length, n_features))
```

```
[ ]: cnn_lstm_model_2 = Sequential()
cnn_lstm_model_2.add(TimeDistributed(Conv1D(filters=256,  padding =␣
 ↪'same',kernel_size=3, activation='relu'),␣
 ↪input_shape=(None,n_length,n_features)))
cnn_lstm_model_2.add(TimeDistributed(Dropout(0.5)))
cnn_lstm_model_2.add(TimeDistributed(Conv1D(filters=64,  padding = 'same'␣
 ↪,kernel_size=3, activation='relu')))
cnn_lstm_model_2.add(TimeDistributed(Dropout(0.5)))
cnn_lstm_model_2.add(TimeDistributed(Flatten()))
cnn_lstm_model_2.add(LSTM(100))
cnn_lstm_model_2.add(Dense(128, activation='relu'))
cnn_lstm_model_2.add(Dense(n_outputs, activation='softmax'))
cnn_lstm_model_2.compile(loss=tf.keras.losses.CategoricalCrossentropy(),␣
 ↪metrics=['accuracy'], optimizer=tf.keras.optimizers.
 ↪Adam(learning_rate=learning_rate))
```

```
[ ]: es = tf.keras.callbacks.EarlyStopping(monitor="val_loss", patience=5,␣
 ↪restore_best_weights=True,)
```

```
checkpointer = ModelCheckpoint(filepath='saved_models/cnn_lstm_classification.
↪hdf5', verbose=1, save_best_only=True)
```

[ ]:
```
tf.config.run_functions_eagerly(True)
```

[ ]:
```
with tf.device('/device:GPU:0'):
  history = cnn_lstm_model_2.fit(X_trainp, training_onehot, batch_size=512,␣
↪epochs=epochs, validation_data=(X_valp, validation_onehot), callbacks=[es,␣
↪checkpointer], verbose=1)
```

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Epoch 1/200
1559/1559 [==============================] - ETA: 0s - loss: 1.5078 - accuracy:
0.3905
Epoch 1: val_loss improved from inf to 1.19245, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 92s 50ms/step - loss: 1.5078 -
accuracy: 0.3905 - val_loss: 1.1925 - val_accuracy: 0.5016
Epoch 2/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.1865 - accuracy:
0.5073
Epoch 2: val_loss improved from 1.19245 to 1.12965, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 80s 51ms/step - loss: 1.1865 -
accuracy: 0.5073 - val_loss: 1.1297 - val_accuracy: 0.5262
Epoch 3/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.1431 - accuracy:
0.5233
Epoch 3: val_loss improved from 1.12965 to 1.11247, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 79s 50ms/step - loss: 1.1430 -
accuracy: 0.5234 - val_loss: 1.1125 - val_accuracy: 0.5324
Epoch 4/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.1172 - accuracy:
0.5336
Epoch 4: val_loss improved from 1.11247 to 1.08792, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 77s 49ms/step - loss: 1.1172 -
accuracy: 0.5336 - val_loss: 1.0879 - val_accuracy: 0.5409
Epoch 5/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.1001 - accuracy:
0.5402
```

```
Epoch 5: val_loss improved from 1.08792 to 1.07411, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 79s 50ms/step - loss: 1.1001 -
accuracy: 0.5402 - val_loss: 1.0741 - val_accuracy: 0.5469
Epoch 6/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0852 - accuracy:
0.5454
Epoch 6: val_loss did not improve from 1.07411
1559/1559 [==============================] - 78s 50ms/step - loss: 1.0851 -
accuracy: 0.5454 - val_loss: 1.0860 - val_accuracy: 0.5428
Epoch 7/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0742 - accuracy:
0.5503
Epoch 7: val_loss improved from 1.07411 to 1.05161, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 79s 50ms/step - loss: 1.0742 -
accuracy: 0.5503 - val_loss: 1.0516 - val_accuracy: 0.5537
Epoch 8/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0645 - accuracy:
0.5540
Epoch 8: val_loss did not improve from 1.05161
1559/1559 [==============================] - 77s 49ms/step - loss: 1.0645 -
accuracy: 0.5540 - val_loss: 1.0585 - val_accuracy: 0.5554
Epoch 9/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0566 - accuracy:
0.5585
Epoch 9: val_loss improved from 1.05161 to 1.03917, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 78s 50ms/step - loss: 1.0566 -
accuracy: 0.5585 - val_loss: 1.0392 - val_accuracy: 0.5624
Epoch 10/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0495 - accuracy:
0.5609
Epoch 10: val_loss improved from 1.03917 to 1.03339, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 77s 50ms/step - loss: 1.0494 -
accuracy: 0.5609 - val_loss: 1.0334 - val_accuracy: 0.5599
Epoch 11/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0376 - accuracy:
0.5653
Epoch 11: val_loss improved from 1.03339 to 1.03225, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 76s 49ms/step - loss: 1.0376 -
accuracy: 0.5653 - val_loss: 1.0322 - val_accuracy: 0.5634
Epoch 12/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0311 - accuracy:
0.5681
Epoch 12: val_loss improved from 1.03225 to 1.02450, saving model to
```

```
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 77s 50ms/step - loss: 1.0311 -
accuracy: 0.5680 - val_loss: 1.0245 - val_accuracy: 0.5672
Epoch 13/200
1559/1559 [==============================] - ETA: 0s - loss: 1.0239 - accuracy:
0.5707
Epoch 13: val_loss improved from 1.02450 to 1.02255, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 78s 50ms/step - loss: 1.0239 -
accuracy: 0.5707 - val_loss: 1.0226 - val_accuracy: 0.5690
Epoch 14/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0155 - accuracy:
0.5739
Epoch 14: val_loss improved from 1.02255 to 1.01164, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 79s 51ms/step - loss: 1.0155 -
accuracy: 0.5739 - val_loss: 1.0116 - val_accuracy: 0.5737
Epoch 15/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0097 - accuracy:
0.5765
Epoch 15: val_loss improved from 1.01164 to 1.01029, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 77s 49ms/step - loss: 1.0097 -
accuracy: 0.5765 - val_loss: 1.0103 - val_accuracy: 0.5738
Epoch 16/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0043 - accuracy:
0.5783
Epoch 16: val_loss improved from 1.01029 to 1.00622, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 77s 49ms/step - loss: 1.0043 -
accuracy: 0.5783 - val_loss: 1.0062 - val_accuracy: 0.5769
Epoch 17/200
1559/1559 [==============================] - ETA: 0s - loss: 1.0011 - accuracy:
0.5810
Epoch 17: val_loss improved from 1.00622 to 0.99874, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 77s 50ms/step - loss: 1.0011 -
accuracy: 0.5810 - val_loss: 0.9987 - val_accuracy: 0.5795
Epoch 18/200
1558/1559 [============================>.] - ETA: 0s - loss: 0.9966 - accuracy:
0.5833
Epoch 18: val_loss did not improve from 0.99874
1559/1559 [==============================] - 75s 48ms/step - loss: 0.9966 -
accuracy: 0.5833 - val_loss: 1.0023 - val_accuracy: 0.5760
Epoch 19/200
1558/1559 [============================>.] - ETA: 0s - loss: 0.9944 - accuracy:
0.5843
Epoch 19: val_loss improved from 0.99874 to 0.99742, saving model to
```

```
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 77s 49ms/step - loss: 0.9945 -
accuracy: 0.5843 - val_loss: 0.9974 - val_accuracy: 0.5813
Epoch 20/200
1558/1559 [============================>.] - ETA: 0s - loss: 0.9917 - accuracy:
0.5859
Epoch 20: val_loss did not improve from 0.99742
1559/1559 [==============================] - 77s 49ms/step - loss: 0.9917 -
accuracy: 0.5859 - val_loss: 0.9998 - val_accuracy: 0.5767
Epoch 21/200
1558/1559 [============================>.] - ETA: 0s - loss: 0.9882 - accuracy:
0.5876
Epoch 21: val_loss improved from 0.99742 to 0.99359, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 78s 50ms/step - loss: 0.9882 -
accuracy: 0.5876 - val_loss: 0.9936 - val_accuracy: 0.5842
Epoch 22/200
1558/1559 [============================>.] - ETA: 0s - loss: 0.9848 - accuracy:
0.5899
Epoch 22: val_loss did not improve from 0.99359
1559/1559 [==============================] - 76s 49ms/step - loss: 0.9848 -
accuracy: 0.5899 - val_loss: 1.0013 - val_accuracy: 0.5815
Epoch 23/200
1558/1559 [============================>.] - ETA: 0s - loss: 0.9823 - accuracy:
0.5914
Epoch 23: val_loss improved from 0.99359 to 0.98730, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 78s 50ms/step - loss: 0.9823 -
accuracy: 0.5914 - val_loss: 0.9873 - val_accuracy: 0.5870
Epoch 24/200
1558/1559 [============================>.] - ETA: 0s - loss: 0.9801 - accuracy:
0.5929
Epoch 24: val_loss did not improve from 0.98730
1559/1559 [==============================] - 76s 49ms/step - loss: 0.9801 -
accuracy: 0.5929 - val_loss: 0.9953 - val_accuracy: 0.5836
Epoch 25/200
1558/1559 [============================>.] - ETA: 0s - loss: 0.9775 - accuracy:
0.5943
Epoch 25: val_loss did not improve from 0.98730
1559/1559 [==============================] - 77s 49ms/step - loss: 0.9775 -
accuracy: 0.5943 - val_loss: 0.9884 - val_accuracy: 0.5854
Epoch 26/200
1558/1559 [============================>.] - ETA: 0s - loss: 0.9753 - accuracy:
0.5960
Epoch 26: val_loss did not improve from 0.98730
1559/1559 [==============================] - 76s 49ms/step - loss: 0.9753 -
accuracy: 0.5960 - val_loss: 0.9949 - val_accuracy: 0.5871
Epoch 27/200
```

```
1558/1559 [============================>.] - ETA: 0s - loss: 0.9734 - accuracy:
0.5973
Epoch 27: val_loss did not improve from 0.98730
1559/1559 [==============================] - 76s 49ms/step - loss: 0.9734 -
accuracy: 0.5973 - val_loss: 0.9900 - val_accuracy: 0.5891
Epoch 28/200
1558/1559 [============================>.] - ETA: 0s - loss: 0.9710 - accuracy:
0.5984
Epoch 28: val_loss improved from 0.98730 to 0.98589, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 77s 50ms/step - loss: 0.9710 -
accuracy: 0.5984 - val_loss: 0.9859 - val_accuracy: 0.5906
Epoch 29/200
1558/1559 [============================>.] - ETA: 0s - loss: 0.9691 - accuracy:
0.5997
Epoch 29: val_loss did not improve from 0.98589
1559/1559 [==============================] - 77s 49ms/step - loss: 0.9691 -
accuracy: 0.5998 - val_loss: 0.9884 - val_accuracy: 0.5895
Epoch 30/200
1559/1559 [==============================] - ETA: 0s - loss: 0.9667 - accuracy:
0.6011
Epoch 30: val_loss did not improve from 0.98589
1559/1559 [==============================] - 77s 50ms/step - loss: 0.9667 -
accuracy: 0.6011 - val_loss: 0.9954 - val_accuracy: 0.5896
Epoch 31/200
1558/1559 [============================>.] - ETA: 0s - loss: 0.9646 - accuracy:
0.6027
Epoch 31: val_loss did not improve from 0.98589
1559/1559 [==============================] - 77s 49ms/step - loss: 0.9646 -
accuracy: 0.6027 - val_loss: 0.9863 - val_accuracy: 0.5914
Epoch 32/200
1558/1559 [============================>.] - ETA: 0s - loss: 0.9629 - accuracy:
0.6039
Epoch 32: val_loss did not improve from 0.98589
1559/1559 [==============================] - 77s 49ms/step - loss: 0.9629 -
accuracy: 0.6039 - val_loss: 0.9866 - val_accuracy: 0.5913
Epoch 33/200
1558/1559 [============================>.] - ETA: 0s - loss: 0.9597 - accuracy:
0.6058
Epoch 33: val_loss improved from 0.98589 to 0.98488, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 77s 50ms/step - loss: 0.9596 -
accuracy: 0.6058 - val_loss: 0.9849 - val_accuracy: 0.5928
Epoch 34/200
1558/1559 [============================>.] - ETA: 0s - loss: 0.9575 - accuracy:
0.6070
Epoch 34: val_loss did not improve from 0.98488
1559/1559 [==============================] - 78s 50ms/step - loss: 0.9576 -
```

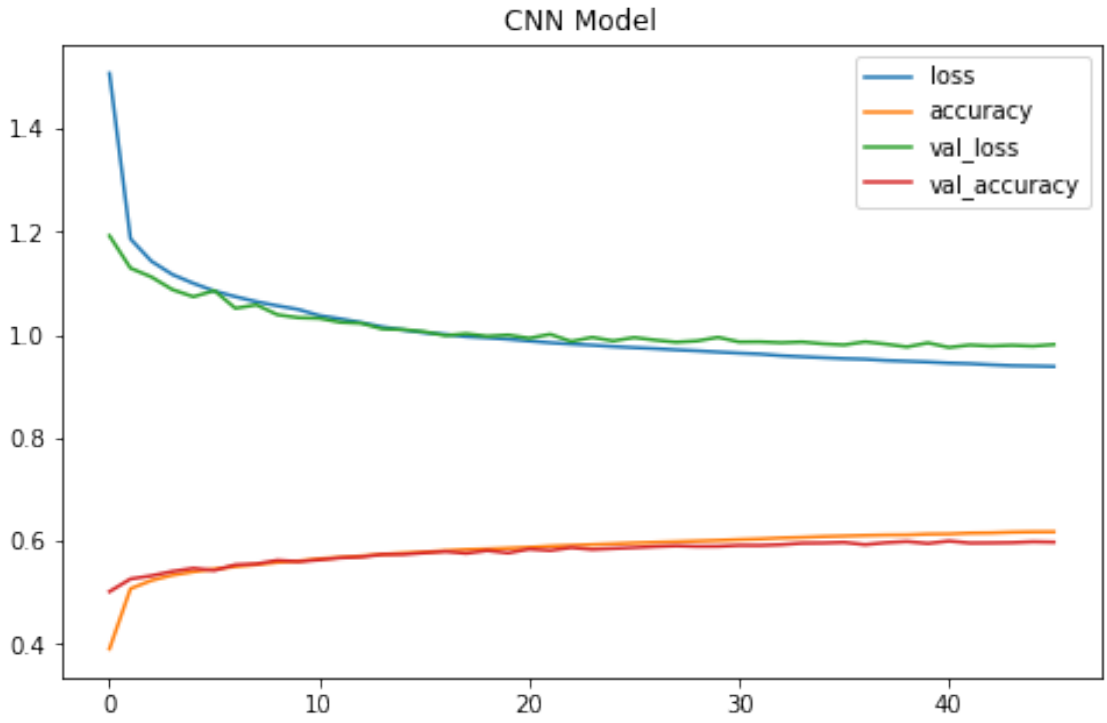
```
accuracy: 0.6069 - val_loss: 0.9863 - val_accuracy: 0.5957
Epoch 35/200
1558/1559 [============================>.] - ETA: 0s - loss: 0.9558 - accuracy:
0.6085
Epoch 35: val_loss improved from 0.98488 to 0.98274, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 77s 50ms/step - loss: 0.9558 -
accuracy: 0.6085 - val_loss: 0.9827 - val_accuracy: 0.5960
Epoch 36/200
1558/1559 [============================>.] - ETA: 0s - loss: 0.9540 - accuracy:
0.6096
Epoch 36: val_loss improved from 0.98274 to 0.98057, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 77s 50ms/step - loss: 0.9540 -
accuracy: 0.6096 - val_loss: 0.9806 - val_accuracy: 0.5973
Epoch 37/200
1558/1559 [============================>.] - ETA: 0s - loss: 0.9528 - accuracy:
0.6107
Epoch 37: val_loss did not improve from 0.98057
1559/1559 [==============================] - 76s 49ms/step - loss: 0.9529 -
accuracy: 0.6107 - val_loss: 0.9867 - val_accuracy: 0.5928
Epoch 38/200
1558/1559 [============================>.] - ETA: 0s - loss: 0.9503 - accuracy:
0.6115
Epoch 38: val_loss did not improve from 0.98057
1559/1559 [==============================] - 76s 49ms/step - loss: 0.9503 -
accuracy: 0.6115 - val_loss: 0.9821 - val_accuracy: 0.5967
Epoch 39/200
1558/1559 [============================>.] - ETA: 0s - loss: 0.9489 - accuracy:
0.6118
Epoch 39: val_loss improved from 0.98057 to 0.97670, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 79s 50ms/step - loss: 0.9489 -
accuracy: 0.6118 - val_loss: 0.9767 - val_accuracy: 0.5989
Epoch 40/200
1558/1559 [============================>.] - ETA: 0s - loss: 0.9474 - accuracy:
0.6133
Epoch 40: val_loss did not improve from 0.97670
1559/1559 [==============================] - 83s 53ms/step - loss: 0.9474 -
accuracy: 0.6133 - val_loss: 0.9846 - val_accuracy: 0.5951
Epoch 41/200
1558/1559 [============================>.] - ETA: 0s - loss: 0.9453 - accuracy:
0.6136
Epoch 41: val_loss improved from 0.97670 to 0.97569, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 79s 51ms/step - loss: 0.9453 -
accuracy: 0.6136 - val_loss: 0.9757 - val_accuracy: 0.5998
Epoch 42/200
```

```
1558/1559 [============================>.] - ETA: 0s - loss: 0.9441 - accuracy:
0.6152
Epoch 42: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 0.9441 -
accuracy: 0.6152 - val_loss: 0.9804 - val_accuracy: 0.5960
Epoch 43/200
1558/1559 [============================>.] - ETA: 0s - loss: 0.9420 - accuracy:
0.6156
Epoch 43: val_loss did not improve from 0.97569
1559/1559 [==============================] - 76s 49ms/step - loss: 0.9420 -
accuracy: 0.6156 - val_loss: 0.9784 - val_accuracy: 0.5962
Epoch 44/200
1558/1559 [============================>.] - ETA: 0s - loss: 0.9403 - accuracy:
0.6170
Epoch 44: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 50ms/step - loss: 0.9403 -
accuracy: 0.6170 - val_loss: 0.9799 - val_accuracy: 0.5967
Epoch 45/200
1558/1559 [============================>.] - ETA: 0s - loss: 0.9396 - accuracy:
0.6177
Epoch 45: val_loss did not improve from 0.97569
1559/1559 [==============================] - 79s 51ms/step - loss: 0.9396 -
accuracy: 0.6177 - val_loss: 0.9783 - val_accuracy: 0.5984
Epoch 46/200
1558/1559 [============================>.] - ETA: 0s - loss: 0.9387 - accuracy:
0.6178
Epoch 46: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 50ms/step - loss: 0.9387 -
accuracy: 0.6178 - val_loss: 0.9813 - val_accuracy: 0.5977
```

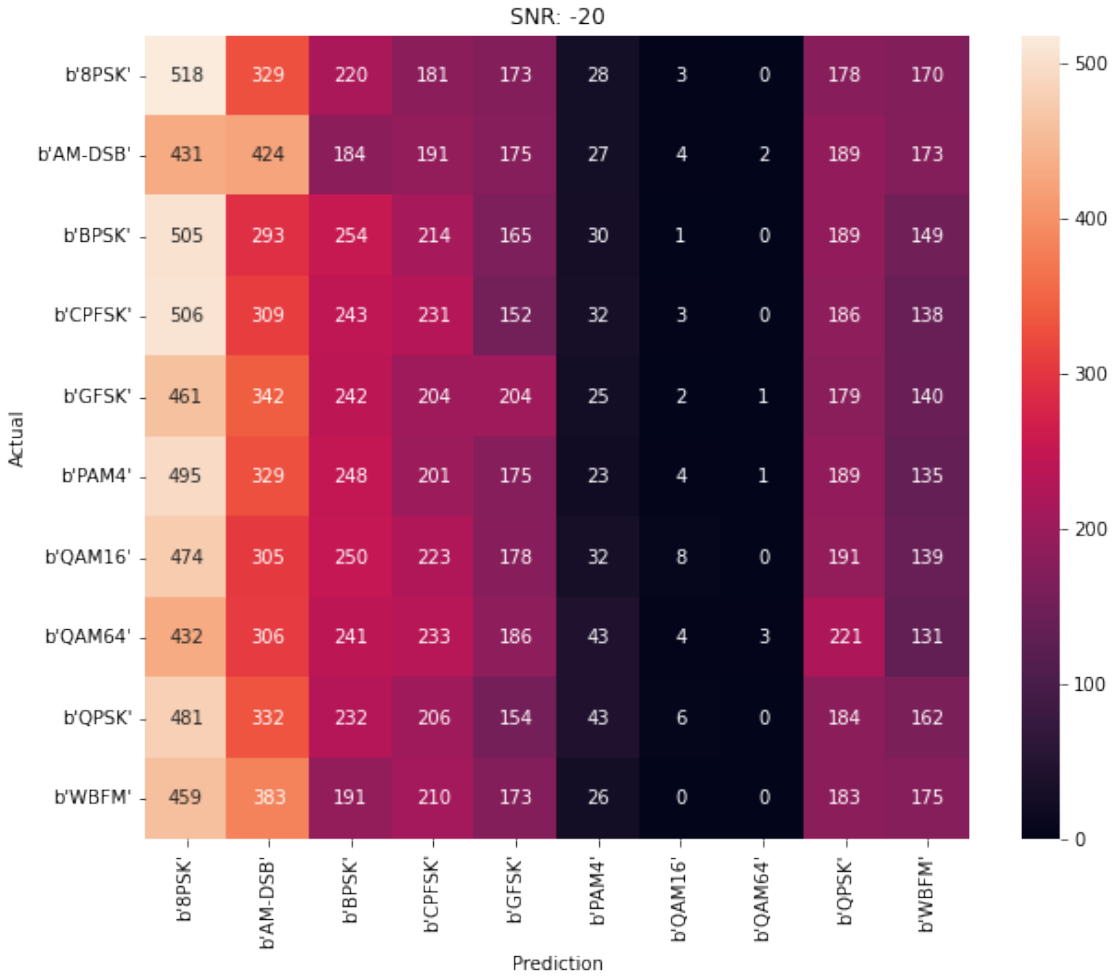
[ ]:
```
'''with tf.device('/device:GPU:0'):
  history = cnn_lstm_model.fit(training_data, training_onehot, batch_size=512,␣
 ↪epochs=1, validation_data=(validation_data, validation_onehot), callbacks=[es,␣
 ↪checkpointer], verbose=1)'''
```

[ ]:
```
"with tf.device('/device:GPU:0'):\n  history = cnn_lstm_model.fit(training_data,
training_onehot, batch_size=512, epochs=1, validation_data=(validation_data,
validation_onehot), callbacks=[es, checkpointer], verbose=1)"
```

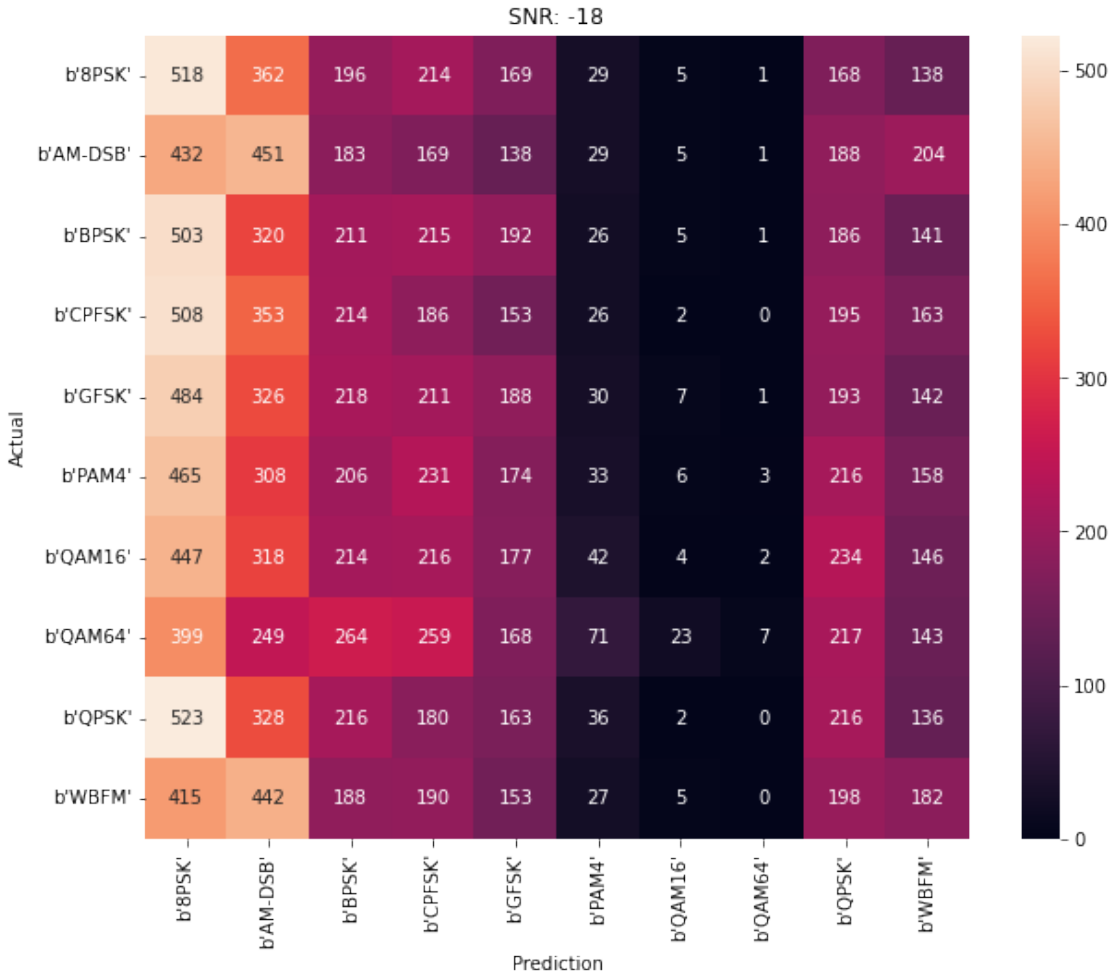
[ ]:
```
plot_model_history(history, 'CNN Model')
model_scoring(cnn_lstm_model_2, X_test, testing_pair_labels)
```



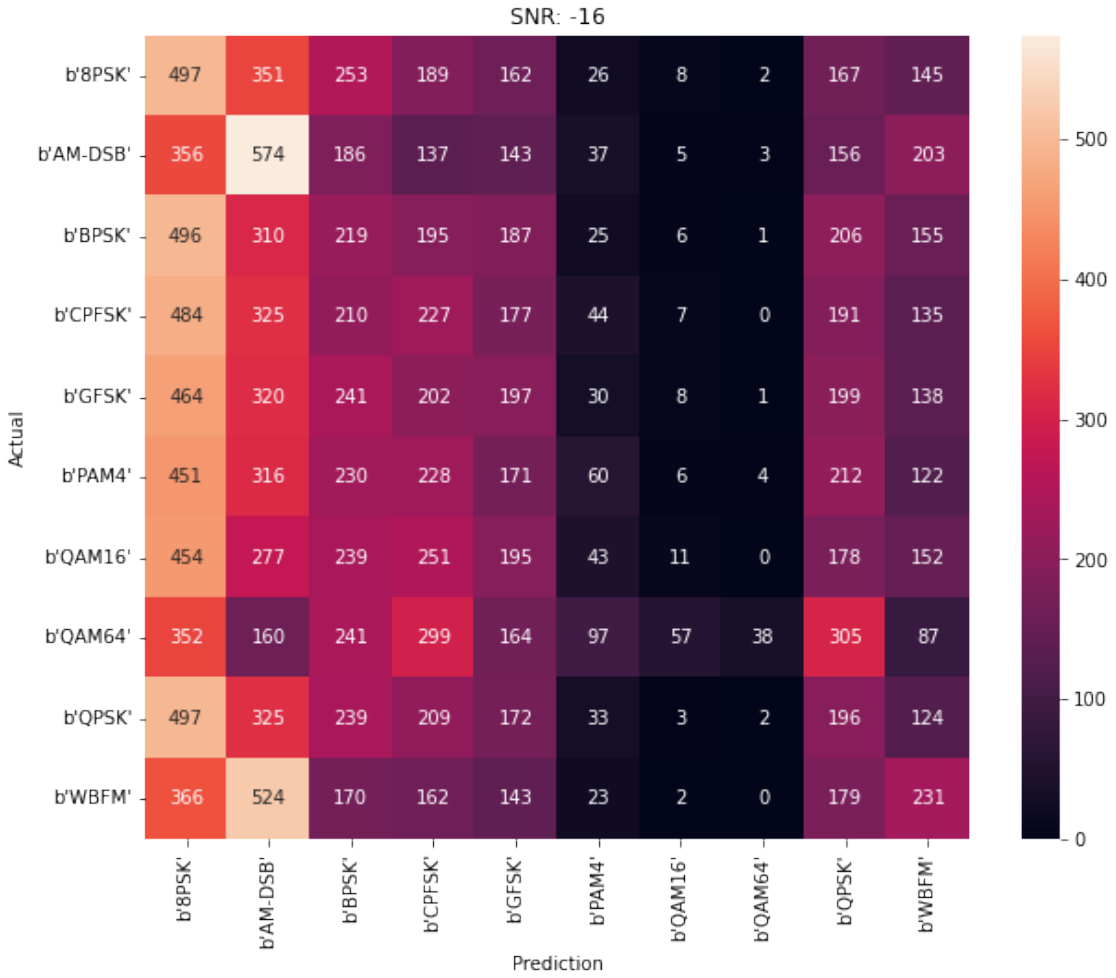

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = -20 is 0.11244444444444444%
```



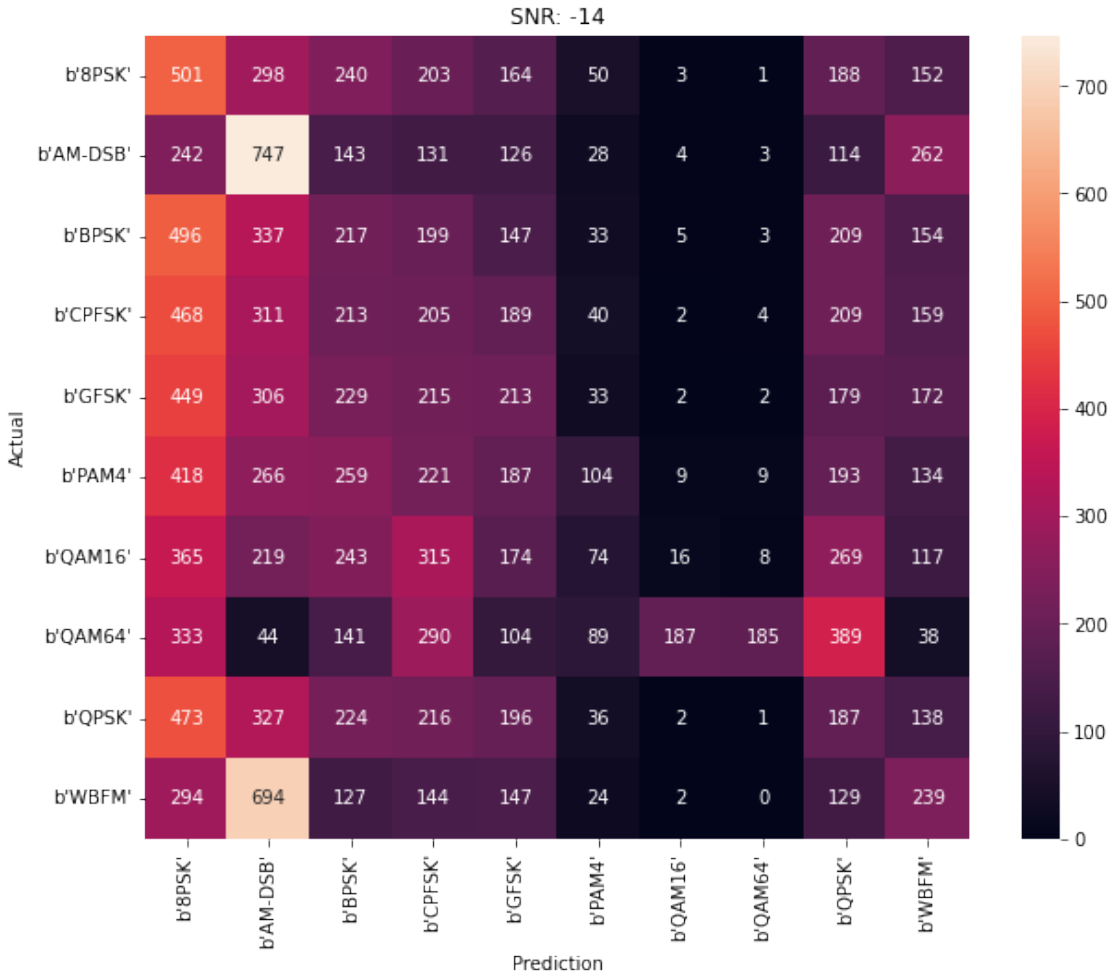

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = -18 is 0.11088888888888888%
```



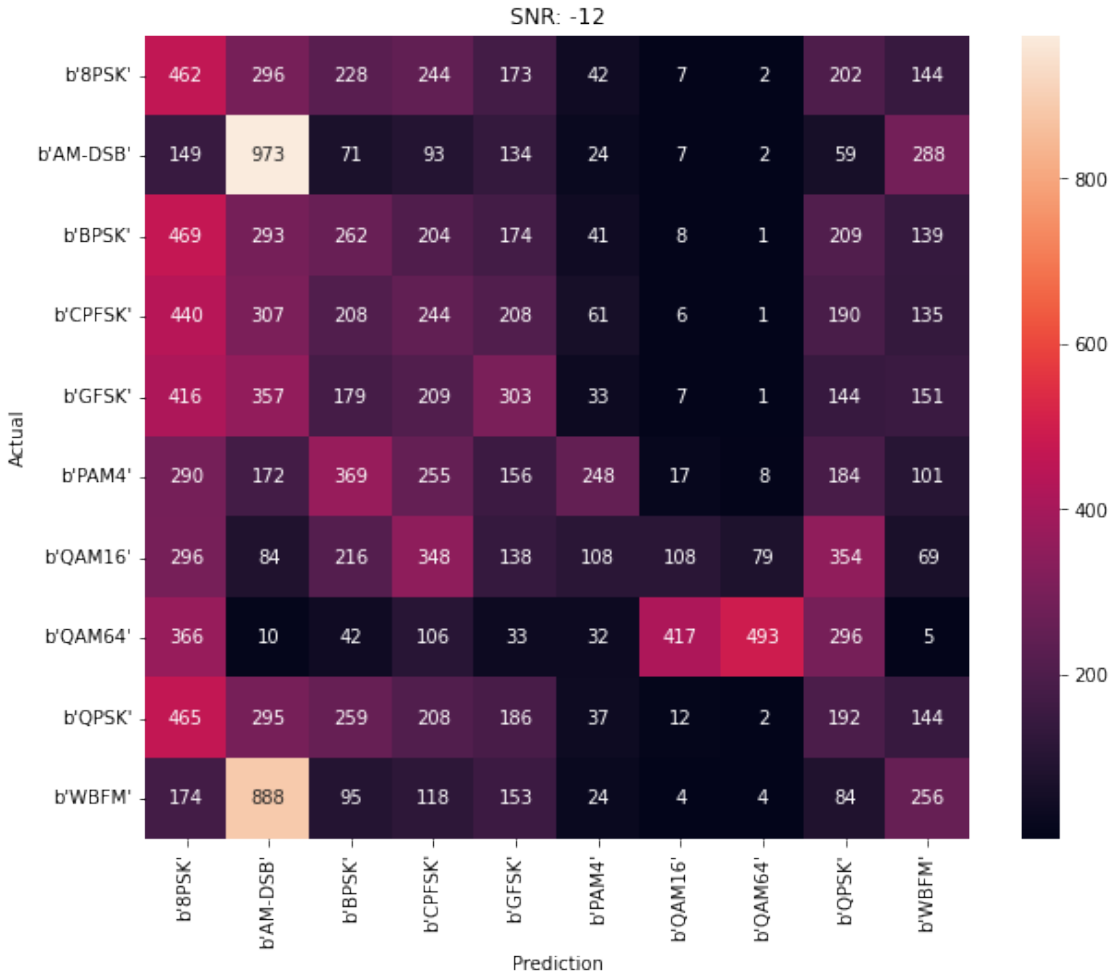

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = -16 is 0.125%
```



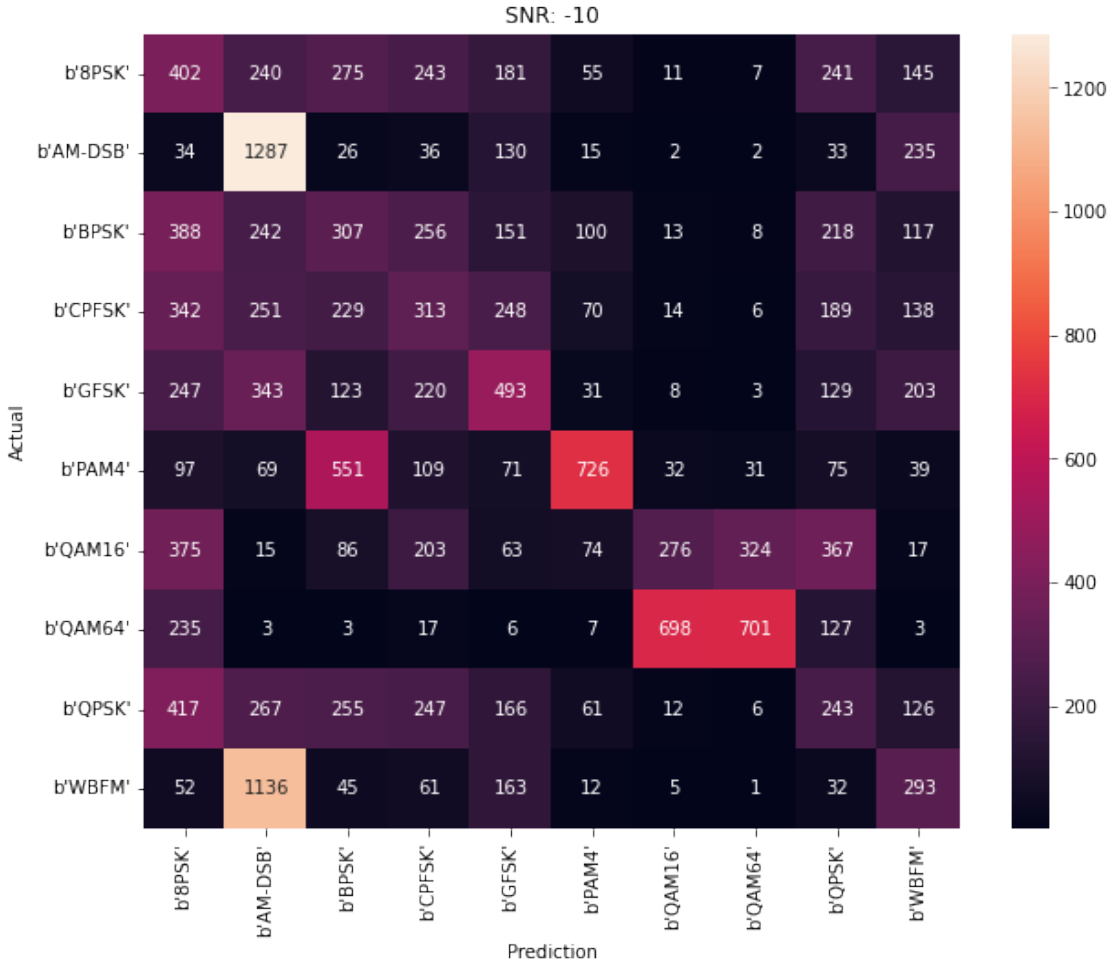

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = -14 is 0.14522222222222222%
```



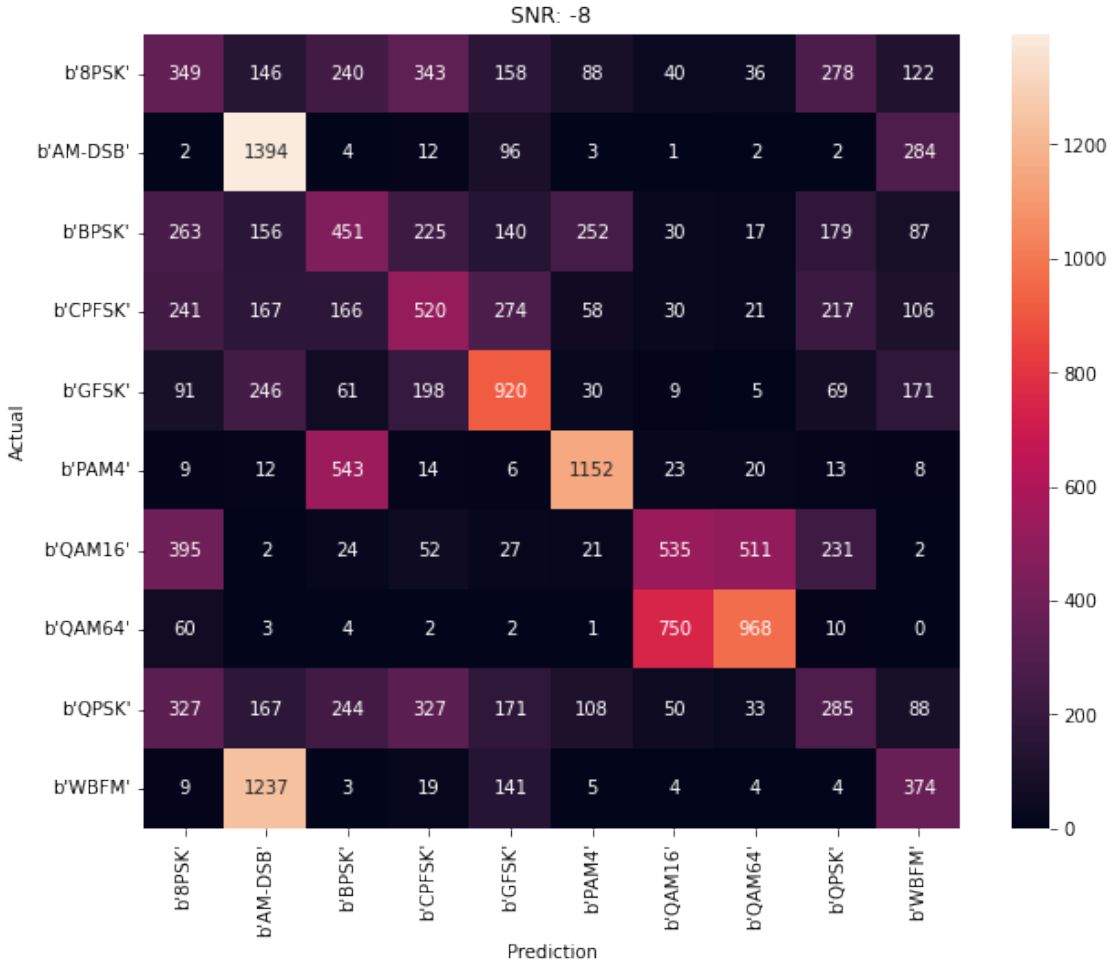

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = -12 is 0.19672222222222221%
```



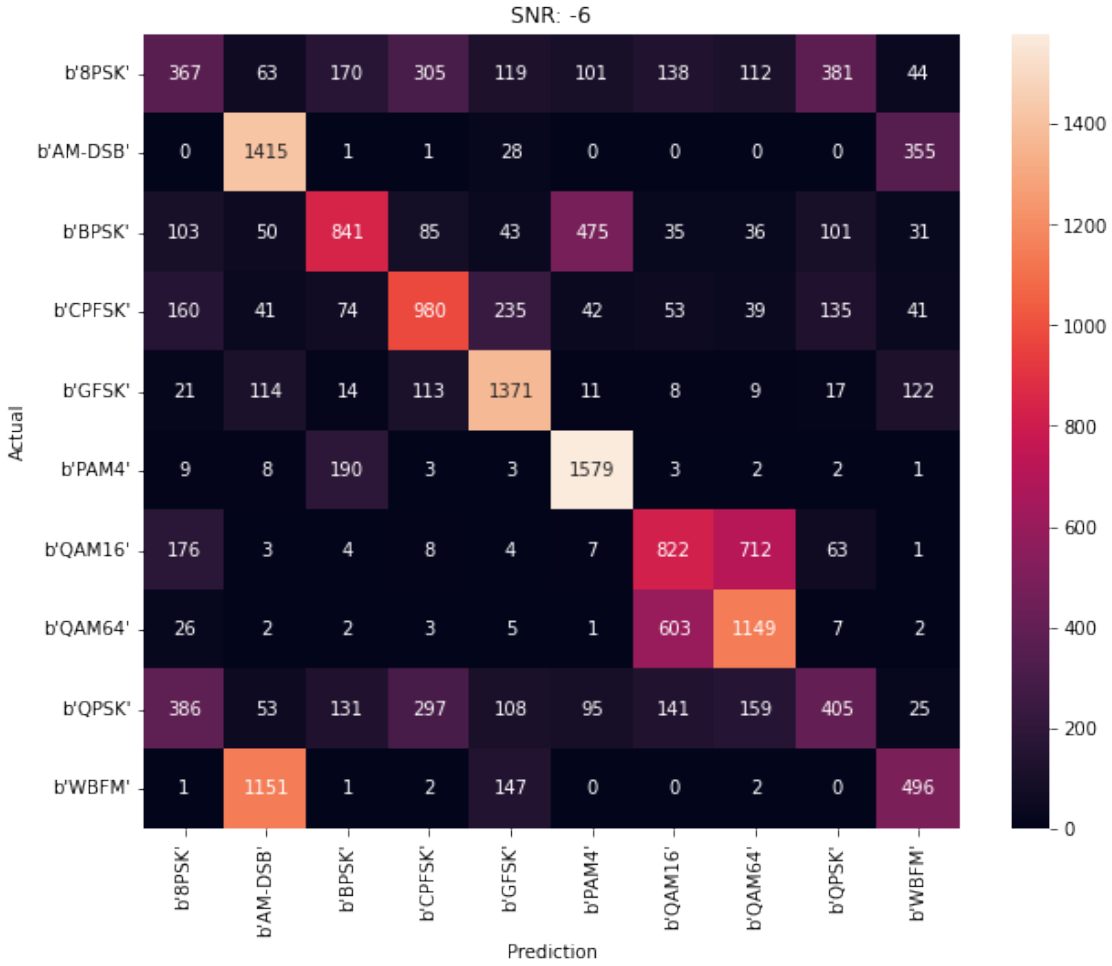

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = -10 is 0.28005555555555556%
```



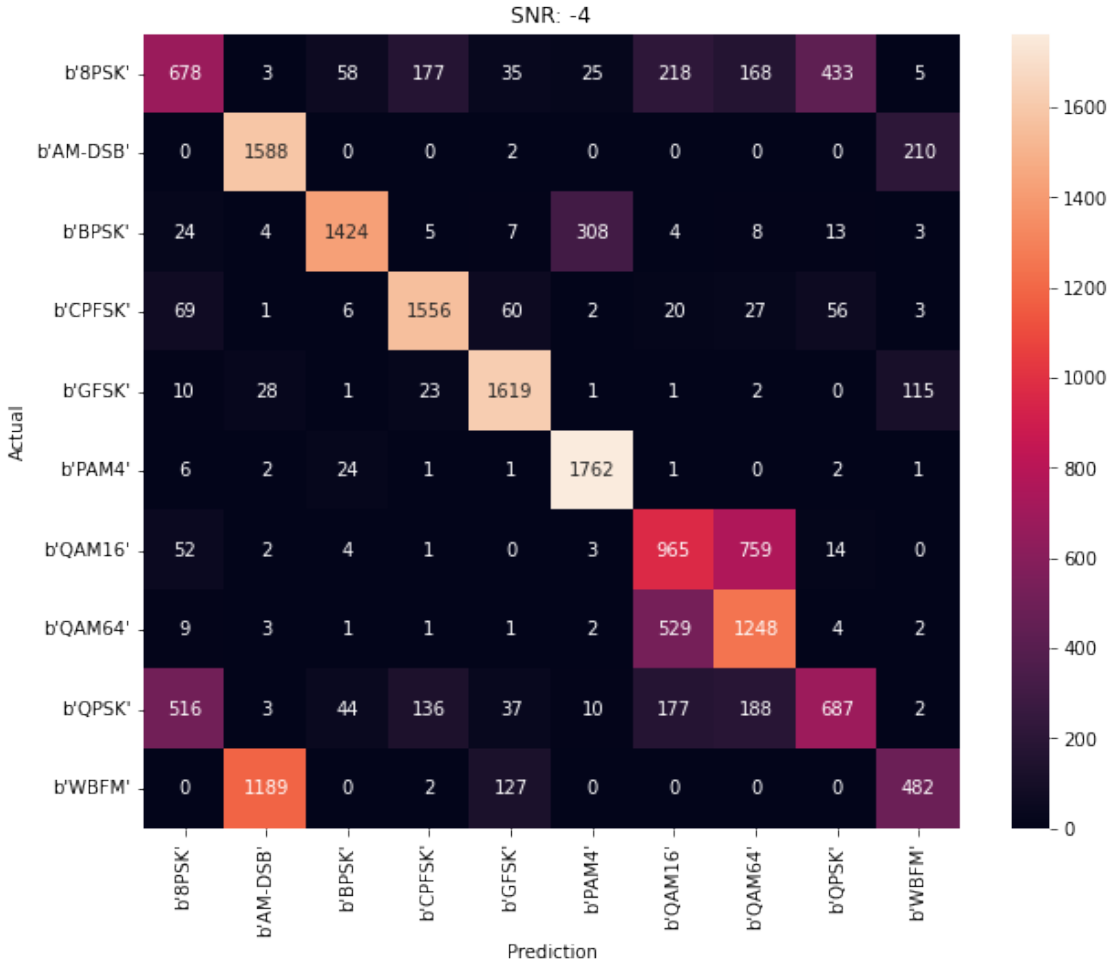

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = -8 is 0.386%
```



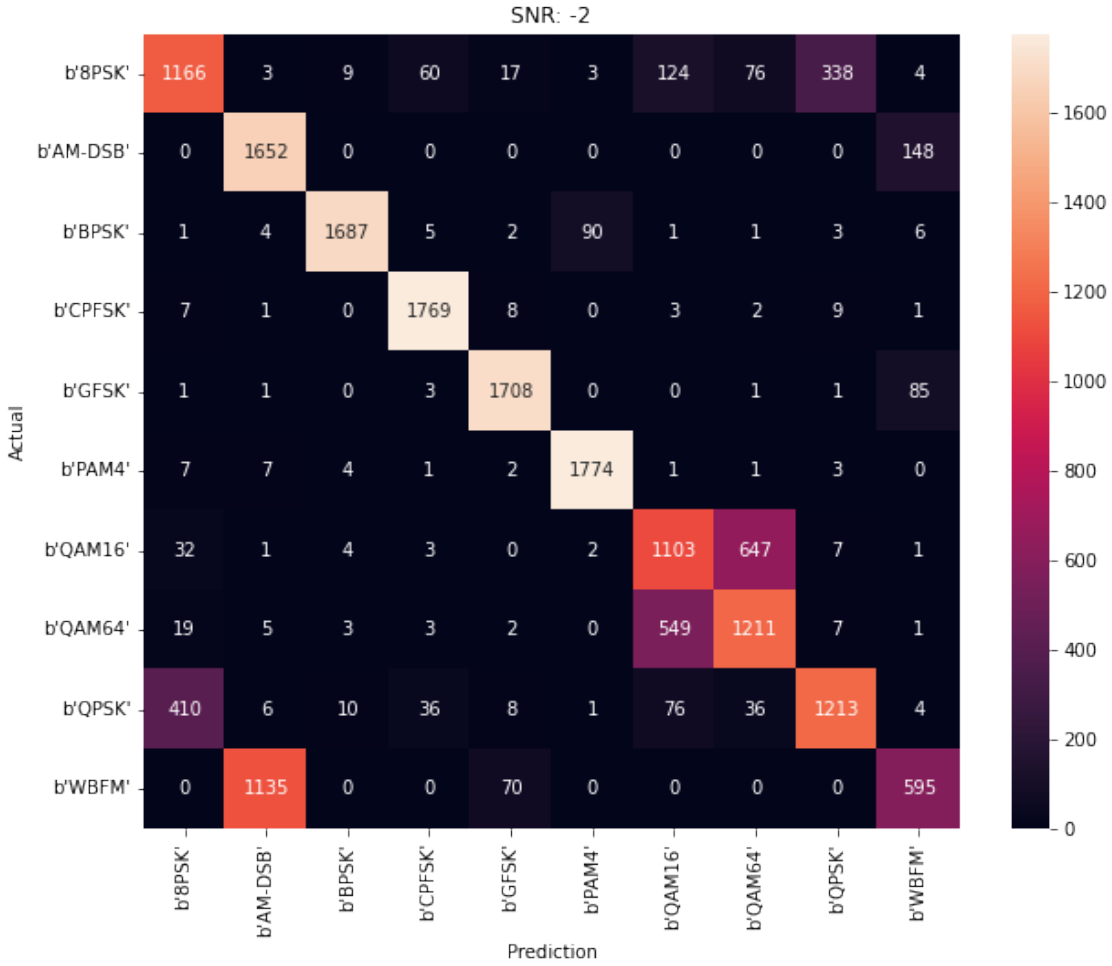

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = -6 is 0.5236111111111111%
```



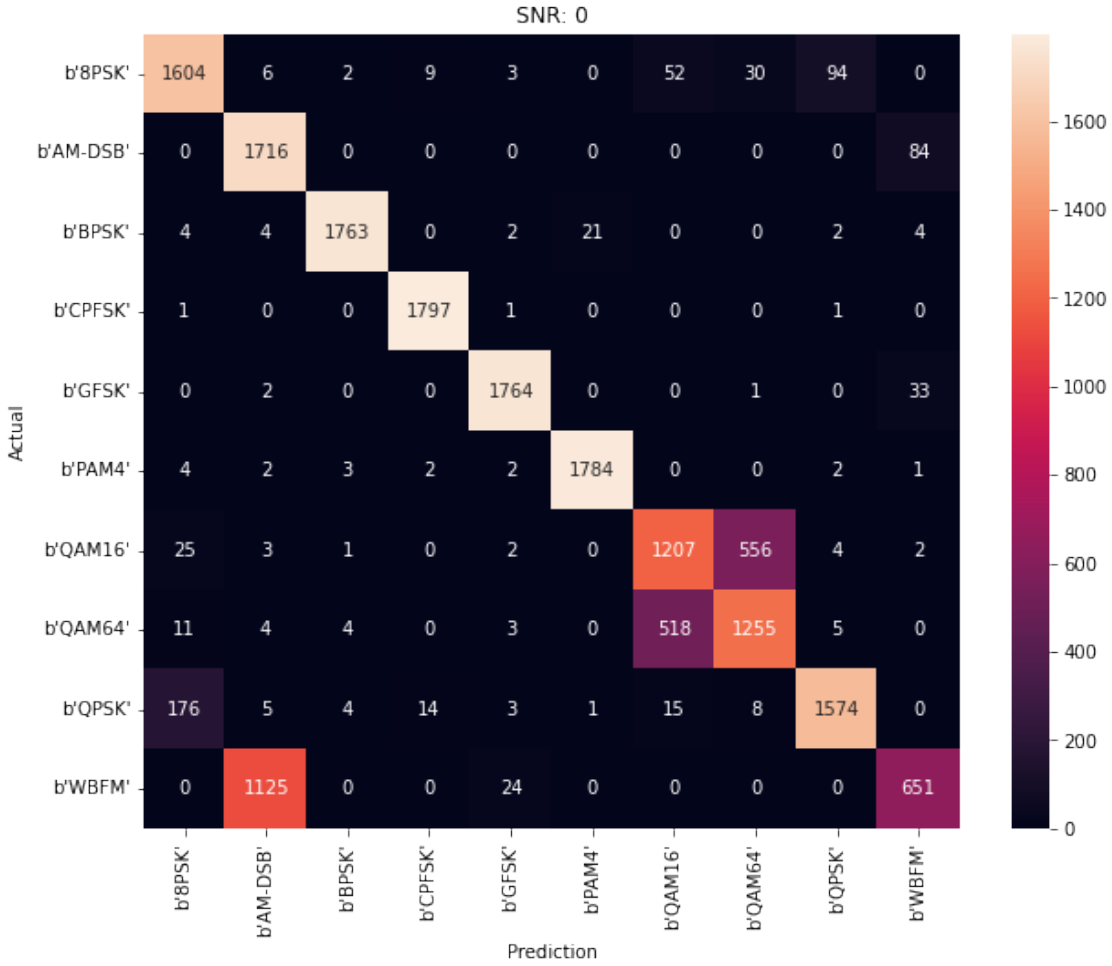

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = -4 is 0.6671666666666667%
```



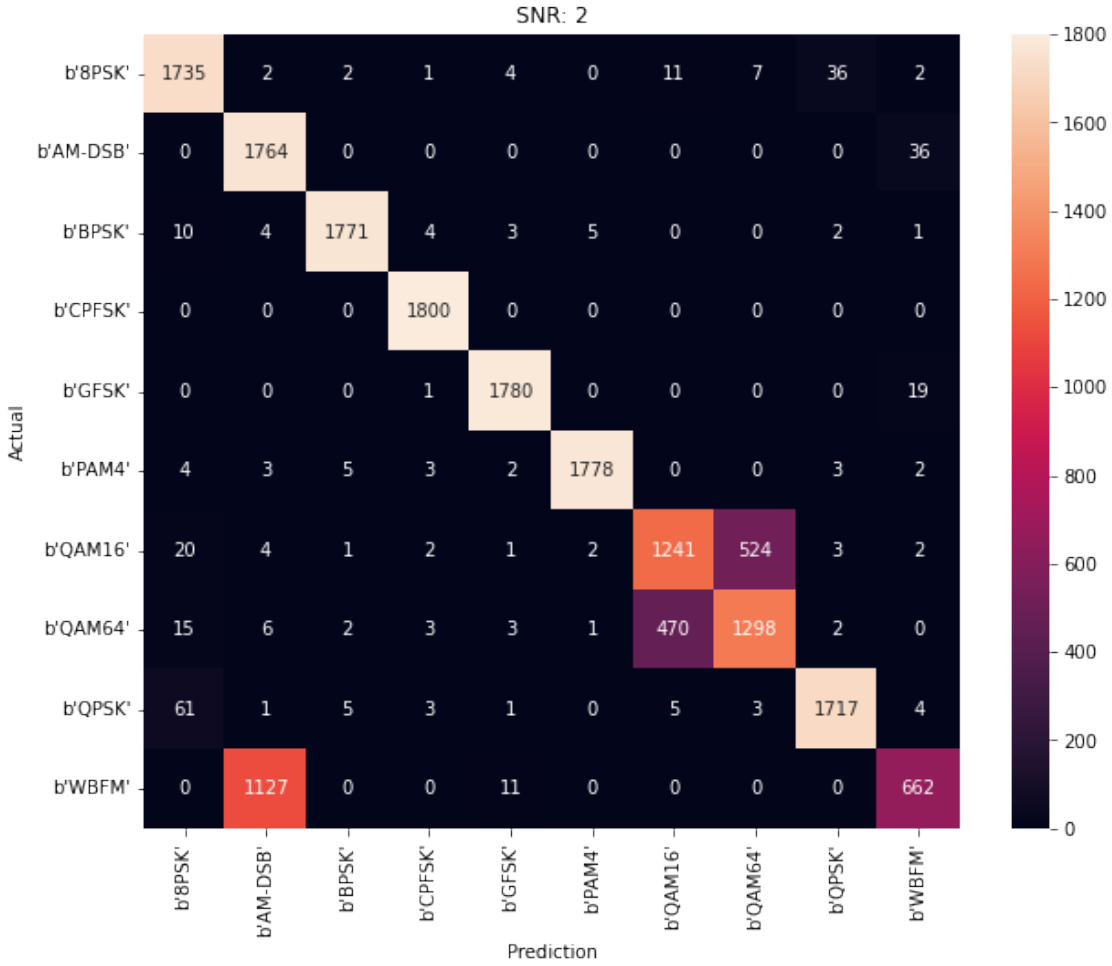

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = -2 is 0.771%
```



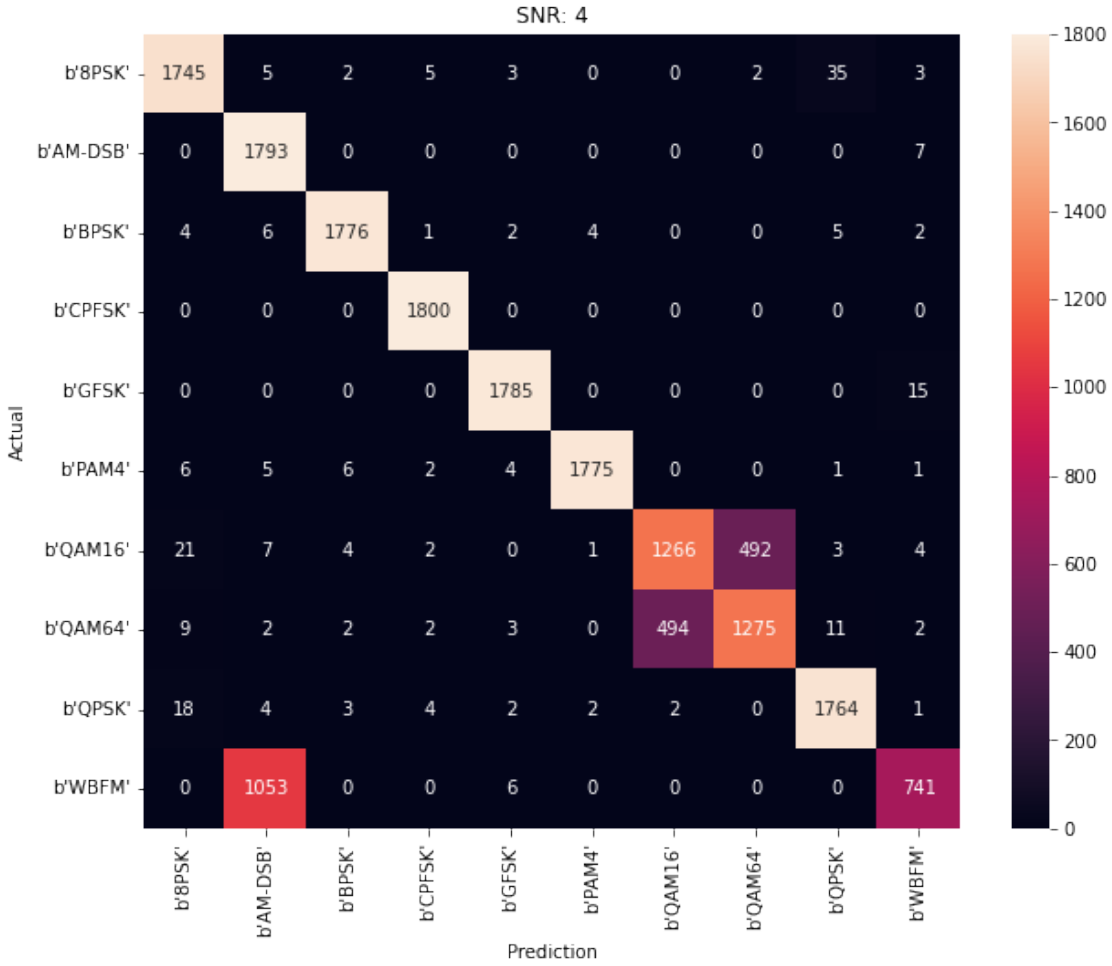

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = 0 is 0.8397222222222223%
```



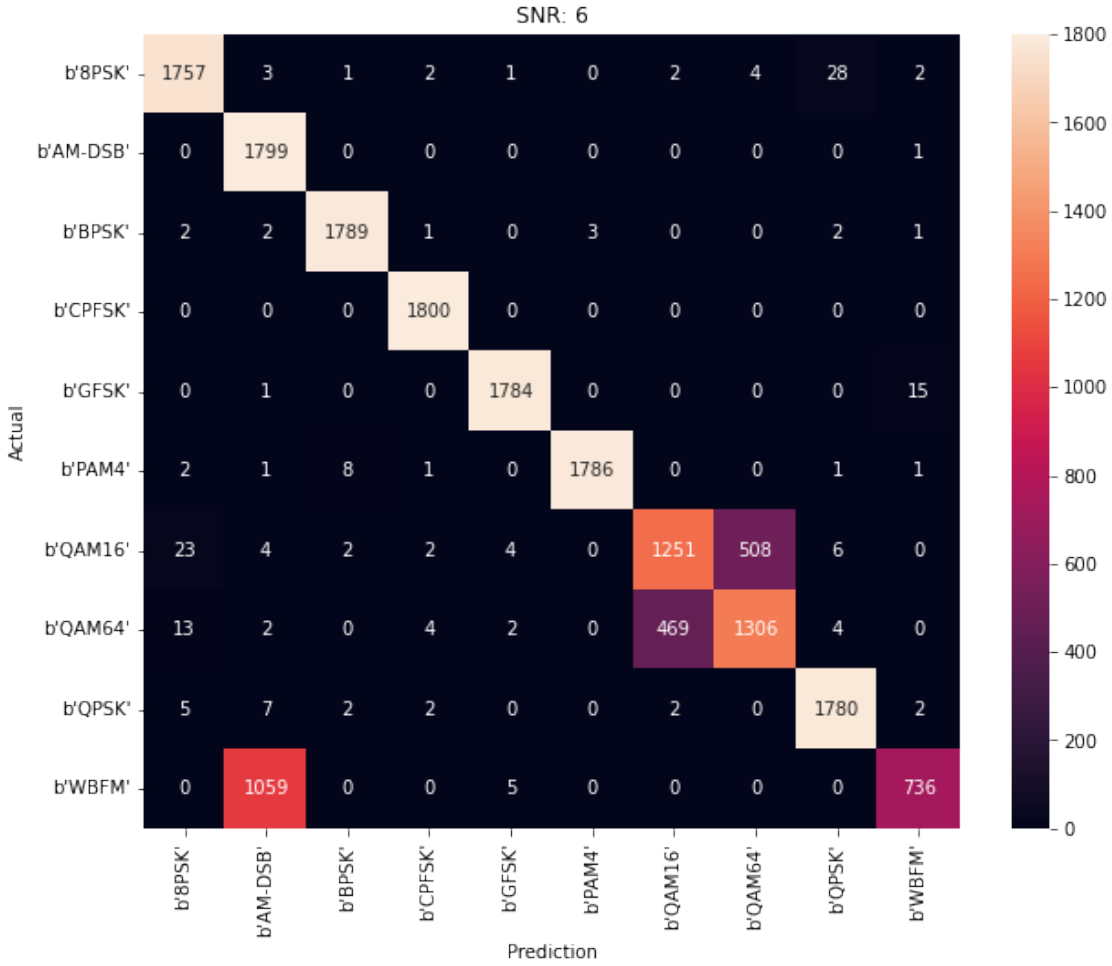

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = 2 is 0.8636666666666667%
```



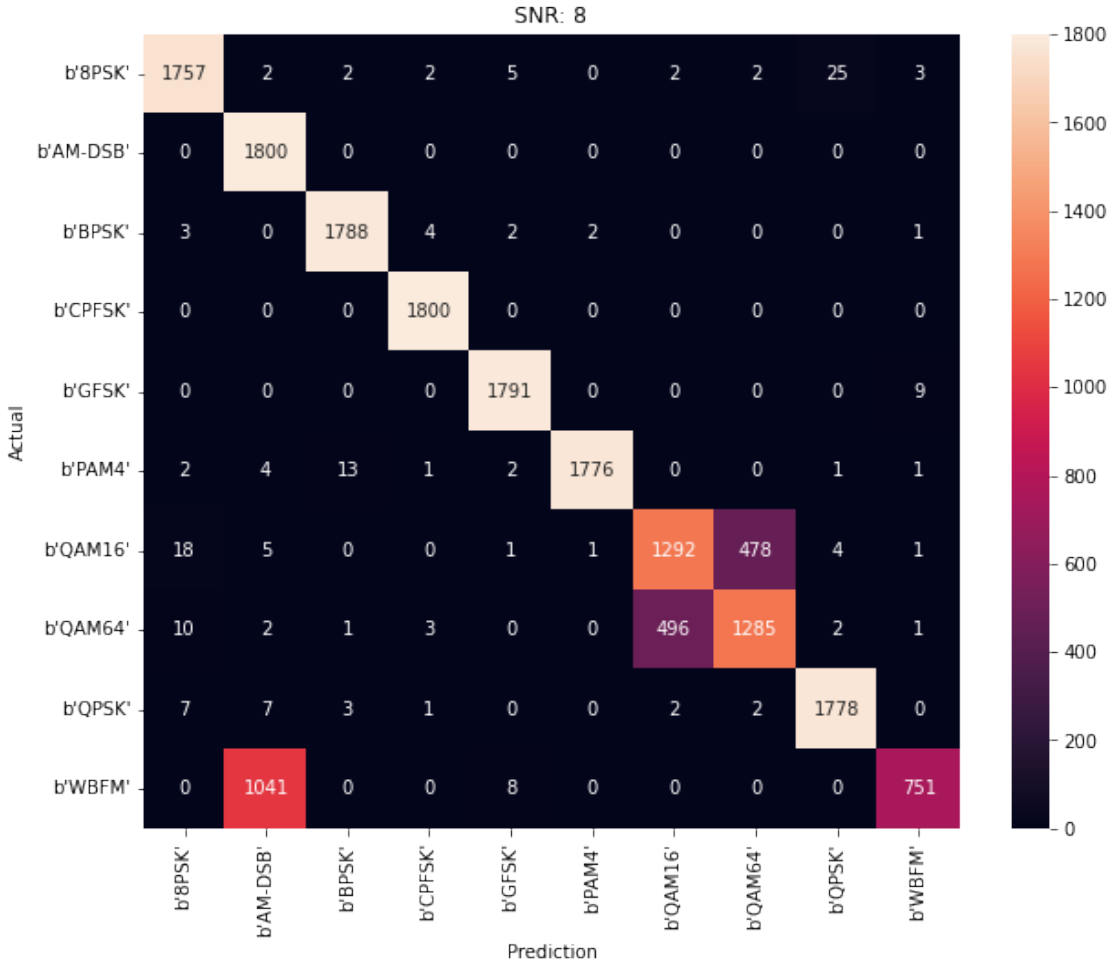

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = 4 is 0.8733333333333333%
```



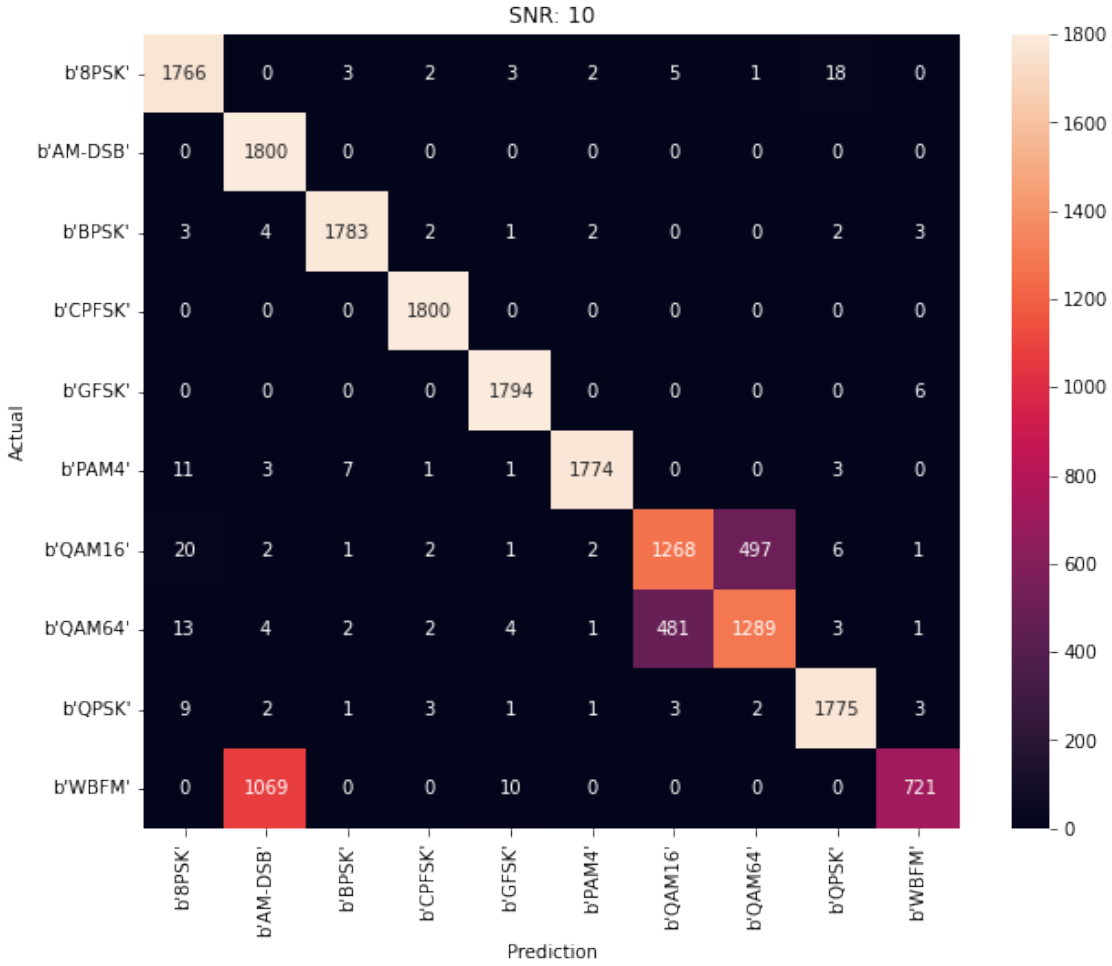

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = 6 is 0.8771111111111111%
```



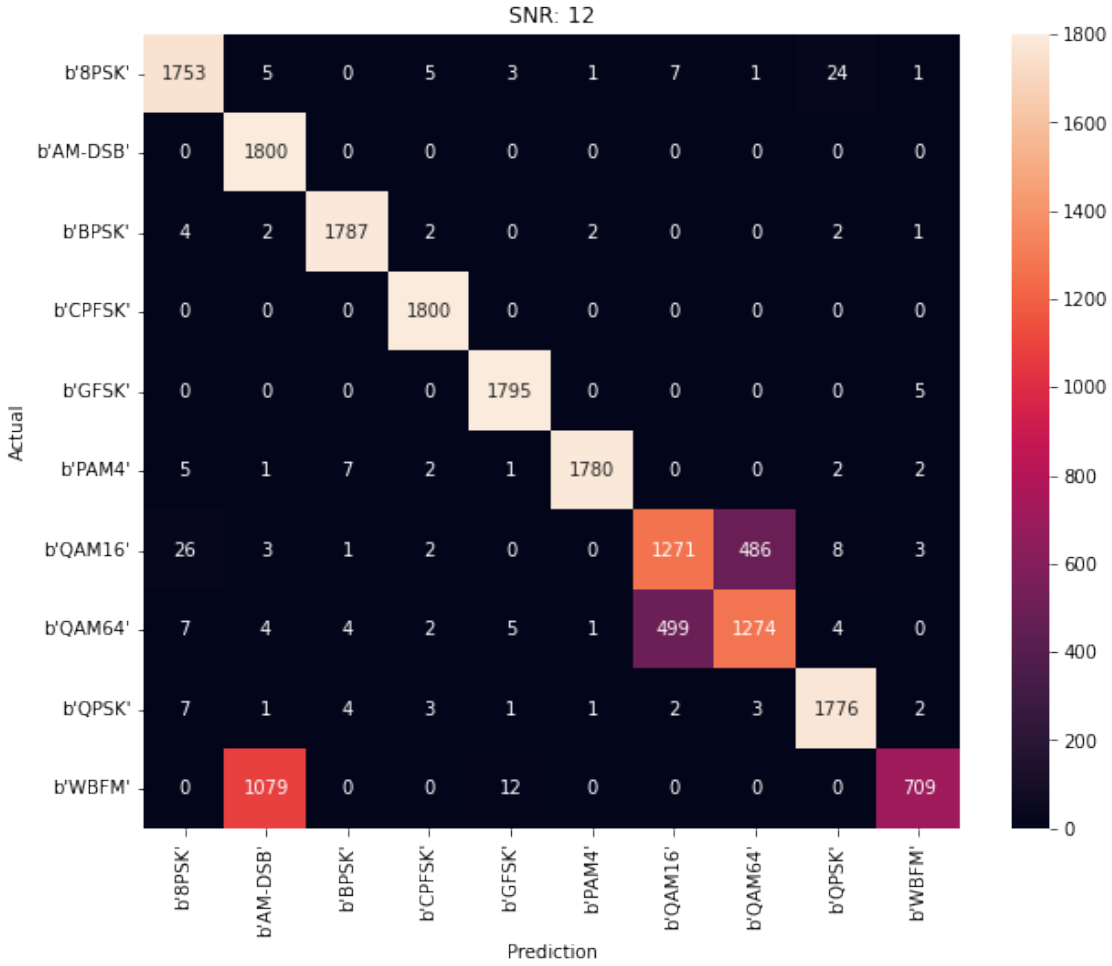

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = 8 is 0.8787777777777778%
```



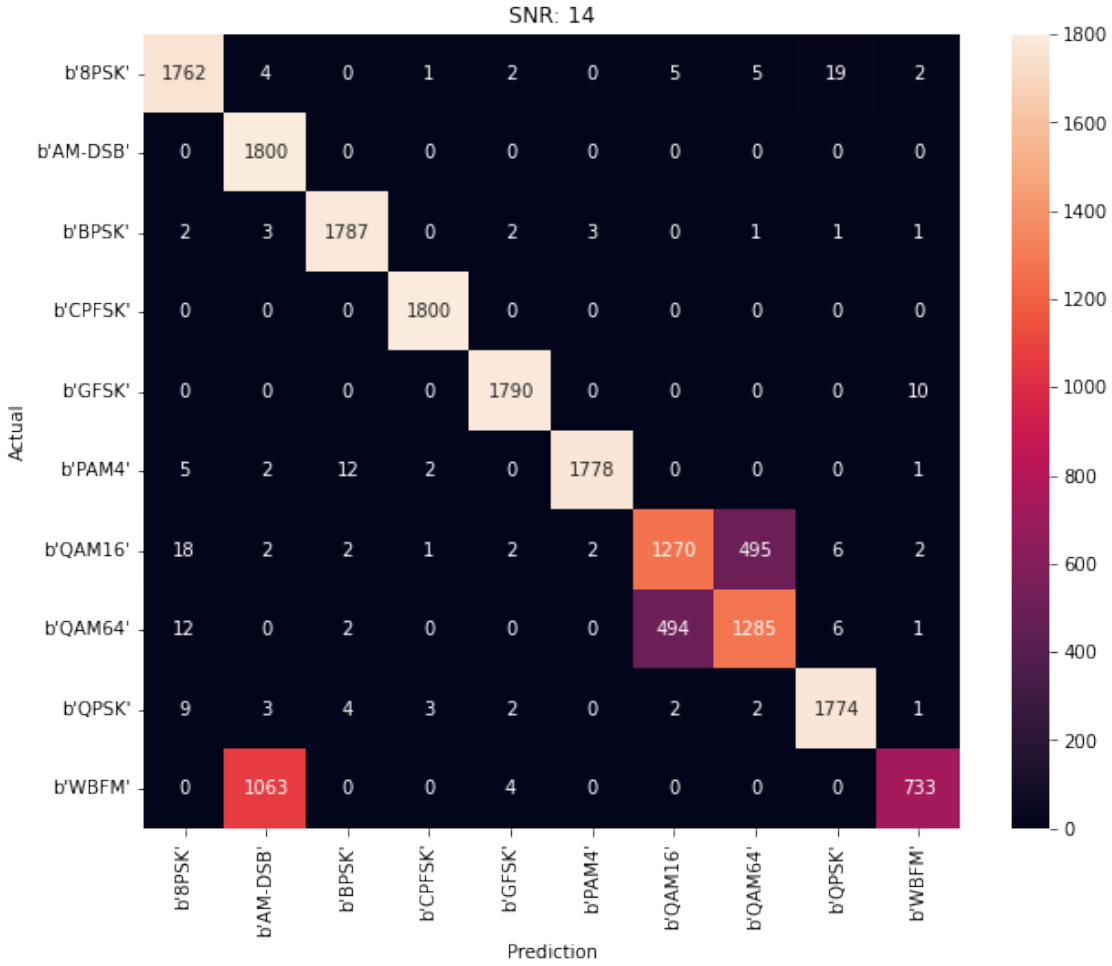

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = 10 is 0.8761111111111111%
```



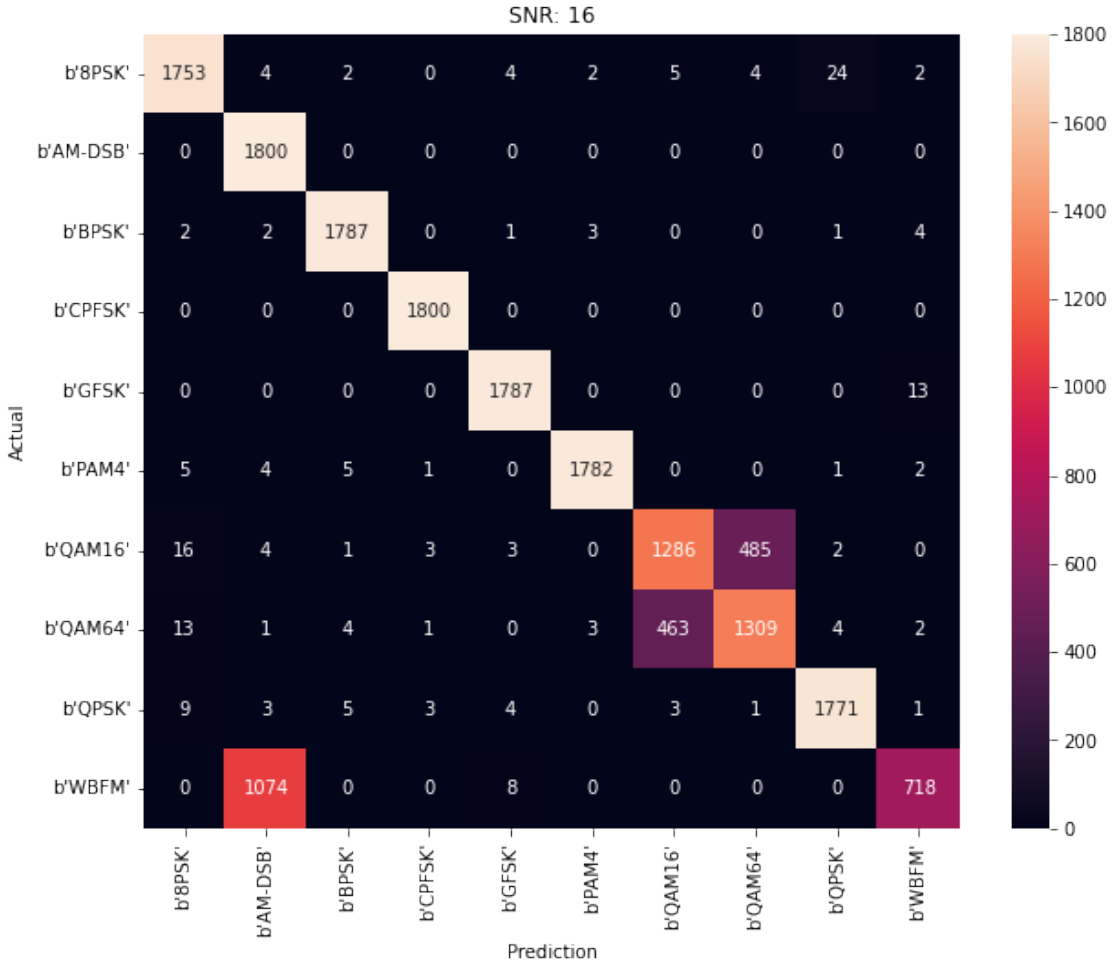

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = 12 is 0.8747222222222222%
```



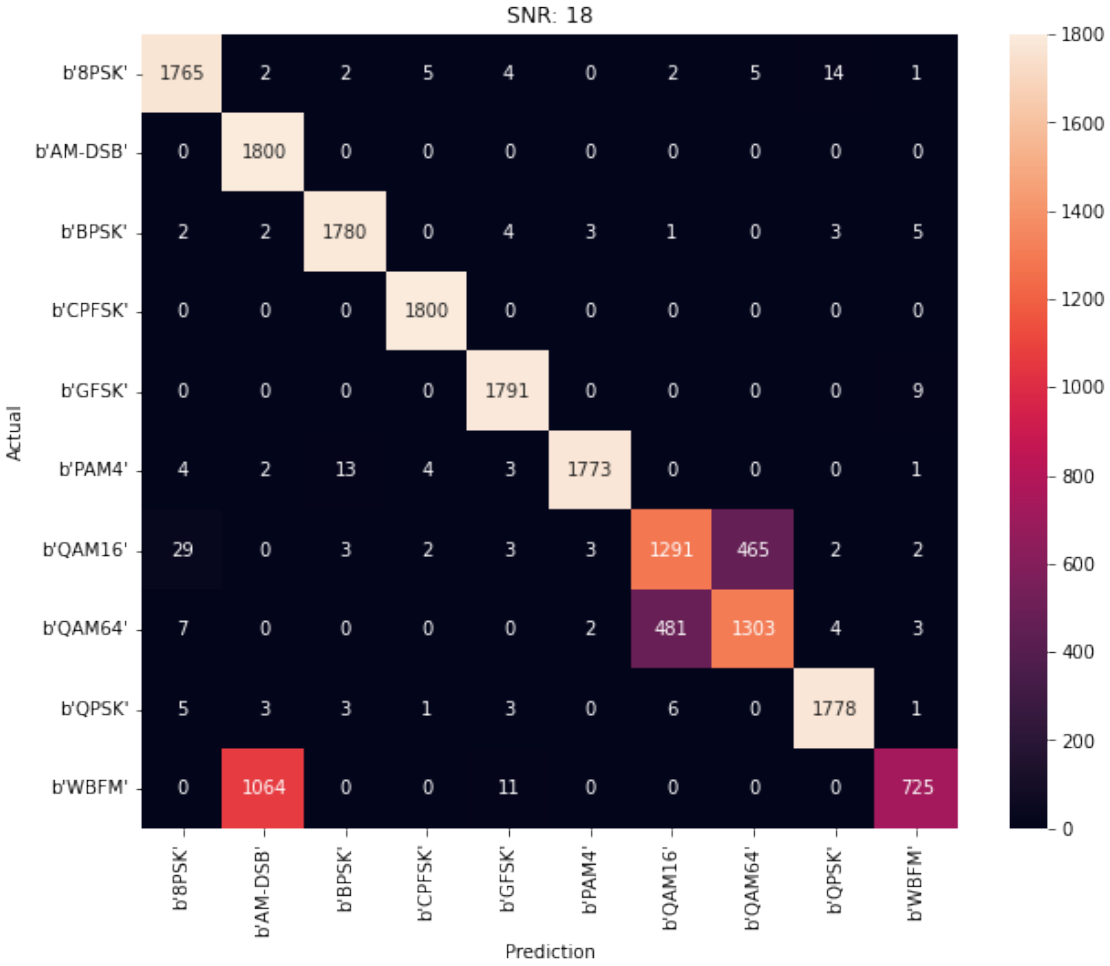

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = 14 is 0.8766111111111111%
```

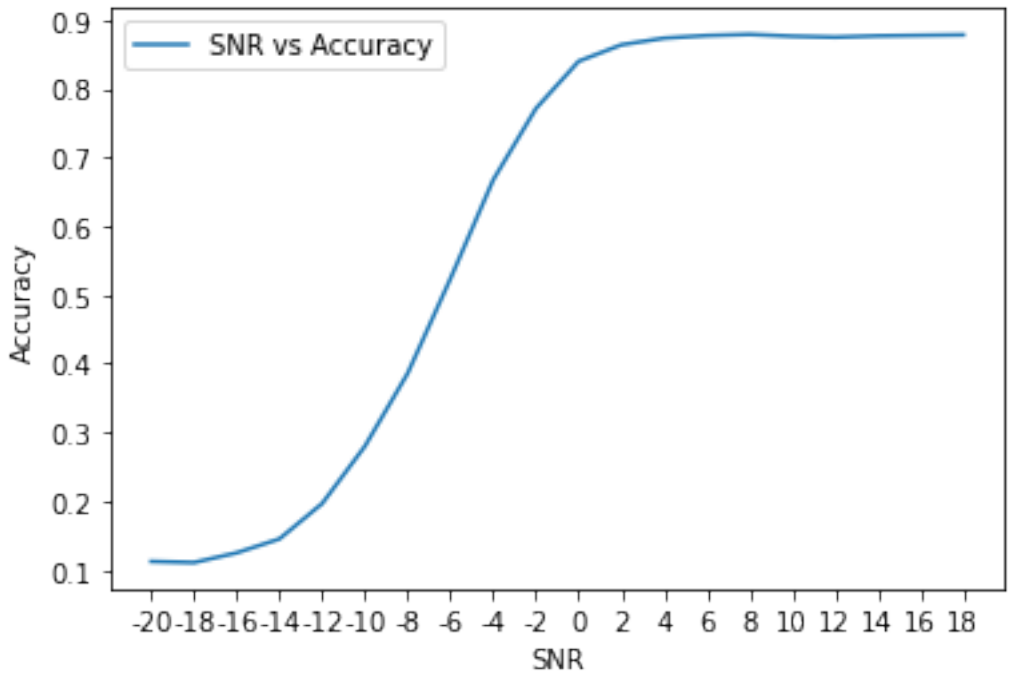



```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = 16 is 0.8773888888888889%
```




```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = 18 is 0.8781111111111111%
```


SNR: 18
b'8PSK' 1765 2 2 5 4 0 2 5 14 1
b'AM-DSB' 0 1800 0 0 0 0 0 0 0 0
b'BPSK' 2 2 1780 0 4 3 1 0 3 5
b'CPFSK' 0 0 0 1800 0 0 0 0 0 0
b'GFSK' 0 0 0 0 1791 0 0 0 0 9
b'PAM4' 4 2 13 4 3 1773 0 0 0 1
b'QAM16' 29 0 3 2 3 3 1291 465 2 2
b'QAM64' 7 0 0 0 0 2 481 1303 4 3
b'QPSK' 5 3 3 1 3 0 6 0 1778 1
b'WBFM' 0 1064 0 0 11 0 0 0 0 725
Actual
b'8PSK' b'AM-DSB' b'BPSK' b'CPFSK' b'GFSK' b'PAM4' b'QAM16' b'QAM64' b'QPSK' b'WBFM'
Prediction
1800 1600 1400 1200 1000 800 600 400 200 0



6 Differentiated Features Space

[17]:
```
fdit_training_data = np.concatenate((normalize(np.gradient(training_data[:,0],␣
 ↪axis = 1)), normalize(np.gradient(training_data[:,1], axis = 1))), axis=1).
 ↪reshape(training_data.shape)
fdit_validation_data = np.concatenate((normalize(np.gradient(validation_data[:
 ↪,0], axis = 1)), normalize(np.gradient(validation_data[:,1], axis = 1))),␣
 ↪axis=1).reshape(validation_data.shape)
fdit_testing_data = np.concatenate((normalize(np.gradient(testing_data[:,0],␣
 ↪axis = 1)), normalize(np.gradient(testing_data[:,1], axis = 1))), axis=1).
 ↪reshape(testing_data.shape)
```

[ ]:
```
print('fdit training data shape:', fdit_training_data.shape)
print('fdit validation data shape:', fdit_validation_data.shape)
print('fdit testing data shape:', fdit_testing_data.shape)
```

```
fdit training data shape: (798000, 2, 128)
fdit validation data shape: (42000, 2, 128)
fdit testing data shape: (360000, 2, 128)
```

[ ]:
```
X_trainp = np.asarray(np.transpose(fdit_training_data, axes=(0,2,1)))
X_valp   = np.asarray(np.transpose(fdit_validation_data  , axes=(0,2,1)))
```

```
n_timesteps, n_features, n_outputs = X_trainp.shape[1], X_trainp.shape[2],␣
 ↪validation_onehot.shape[1]
n_steps, n_length = 4, 32
X_trainp = X_trainp.reshape((X_trainp.shape[0], n_steps, n_length, n_features))
X_valp = X_valp.reshape((X_valp.shape[0], n_steps, n_length, n_features))
```

[ ]:
```
X_test = np.asarray(np.transpose(fdit_testing_data, axes=(0,2,1)))
n_timesteps, n_features, n_outputs = X_test.shape[1], X_test.shape[2],␣
 ↪validation_onehot.shape[1]
n_steps, n_length = 4, 32
X_test = X_test.reshape((X_test.shape[0], n_steps, n_length, n_features))
```

[ ]:
```
es = tf.keras.callbacks.EarlyStopping(monitor="val_loss", patience=5,␣
 ↪restore_best_weights=True,)
checkpointer = ModelCheckpoint(filepath='saved_models/cnn_lstm_classification.
 ↪hdf5', verbose=1, save_best_only=True)
```

[ ]:
```
cnn_lstm_model_2 = Sequential()
cnn_lstm_model_2.add(TimeDistributed(Conv1D(filters=256,  padding =␣
 ↪'same',kernel_size=3, activation='relu'),␣
 ↪input_shape=(None,n_length,n_features)))
cnn_lstm_model_2.add(TimeDistributed(Dropout(0.5)))
cnn_lstm_model_2.add(TimeDistributed(Conv1D(filters=64,  padding = 'same'␣
 ↪,kernel_size=3, activation='relu')))
cnn_lstm_model_2.add(TimeDistributed(Dropout(0.5)))
cnn_lstm_model_2.add(TimeDistributed(Flatten()))
cnn_lstm_model_2.add(LSTM(100))
cnn_lstm_model_2.add(Dense(128, activation='relu'))
cnn_lstm_model_2.add(Dense(n_outputs, activation='softmax'))
cnn_lstm_model_2.compile(loss=tf.keras.losses.CategoricalCrossentropy(),␣
 ↪metrics=['accuracy'], optimizer=tf.keras.optimizers.
 ↪Adam(learning_rate=learning_rate))
```

[ ]:
```
with tf.device('/device:GPU:0'):
  history = cnn_lstm_model_2.fit(X_trainp, training_onehot, batch_size=512,␣
 ↪epochs=epochs, validation_data=(X_valp, validation_onehot), callbacks=[es,␣
 ↪checkpointer], verbose=1)
```

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Epoch 1/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.7376 - accuracy:
```

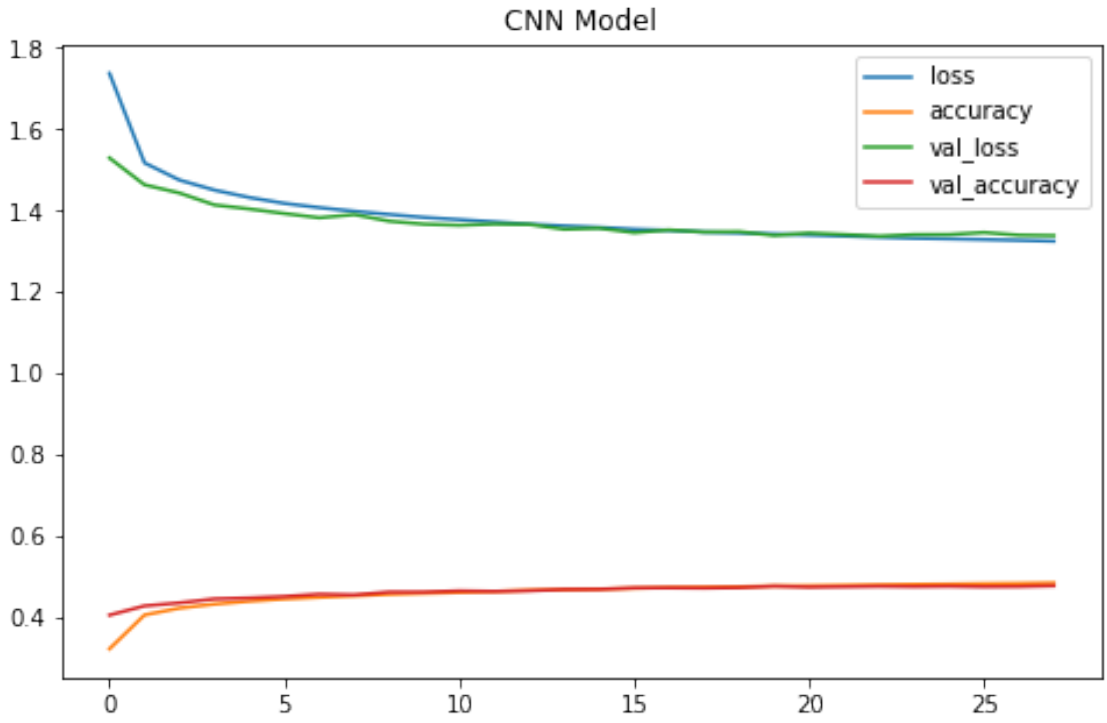
```
0.3233
Epoch 1: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.7375 -
accuracy: 0.3233 - val_loss: 1.5296 - val_accuracy: 0.4062
Epoch 2/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.5176 - accuracy:
0.4066
Epoch 2: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.5176 -
accuracy: 0.4066 - val_loss: 1.4637 - val_accuracy: 0.4289
Epoch 3/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.4754 - accuracy:
0.4234
Epoch 3: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.4753 -
accuracy: 0.4234 - val_loss: 1.4435 - val_accuracy: 0.4363
Epoch 4/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.4506 - accuracy:
0.4332
Epoch 4: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 49ms/step - loss: 1.4506 -
accuracy: 0.4332 - val_loss: 1.4142 - val_accuracy: 0.4461
Epoch 5/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.4324 - accuracy:
0.4404
Epoch 5: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 49ms/step - loss: 1.4324 -
accuracy: 0.4405 - val_loss: 1.4042 - val_accuracy: 0.4484
Epoch 6/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.4180 - accuracy:
0.4460
Epoch 6: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.4180 -
accuracy: 0.4460 - val_loss: 1.3930 - val_accuracy: 0.4520
Epoch 7/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.4076 - accuracy:
0.4500
Epoch 7: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.4076 -
accuracy: 0.4499 - val_loss: 1.3827 - val_accuracy: 0.4576
Epoch 8/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.3986 - accuracy:
0.4536
Epoch 8: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.3986 -
accuracy: 0.4536 - val_loss: 1.3904 - val_accuracy: 0.4559
Epoch 9/200
1559/1559 [==============================] - ETA: 0s - loss: 1.3908 - accuracy:
```

```
0.4574
Epoch 9: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 50ms/step - loss: 1.3908 -
accuracy: 0.4574 - val_loss: 1.3743 - val_accuracy: 0.4629
Epoch 10/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.3835 - accuracy:
0.4599
Epoch 10: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.3835 -
accuracy: 0.4599 - val_loss: 1.3669 - val_accuracy: 0.4628
Epoch 11/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.3781 - accuracy:
0.4620
Epoch 11: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.3781 -
accuracy: 0.4620 - val_loss: 1.3643 - val_accuracy: 0.4658
Epoch 12/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.3726 - accuracy:
0.4642
Epoch 12: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 49ms/step - loss: 1.3726 -
accuracy: 0.4642 - val_loss: 1.3672 - val_accuracy: 0.4645
Epoch 13/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.3668 - accuracy:
0.4673
Epoch 13: val_loss did not improve from 0.97569
1559/1559 [==============================] - 79s 50ms/step - loss: 1.3668 -
accuracy: 0.4673 - val_loss: 1.3664 - val_accuracy: 0.4661
Epoch 14/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.3627 - accuracy:
0.4686
Epoch 14: val_loss did not improve from 0.97569
1559/1559 [==============================] - 76s 49ms/step - loss: 1.3626 -
accuracy: 0.4686 - val_loss: 1.3544 - val_accuracy: 0.4692
Epoch 15/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.3590 - accuracy:
0.4702
Epoch 15: val_loss did not improve from 0.97569
1559/1559 [==============================] - 79s 50ms/step - loss: 1.3590 -
accuracy: 0.4702 - val_loss: 1.3569 - val_accuracy: 0.4696
Epoch 16/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.3544 - accuracy:
0.4723
Epoch 16: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 49ms/step - loss: 1.3544 -
accuracy: 0.4723 - val_loss: 1.3461 - val_accuracy: 0.4730
Epoch 17/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.3518 - accuracy:
```

```
0.4742
Epoch 17: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 49ms/step - loss: 1.3519 -
accuracy: 0.4742 - val_loss: 1.3528 - val_accuracy: 0.4732
Epoch 18/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.3475 - accuracy:
0.4751
Epoch 18: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.3475 -
accuracy: 0.4751 - val_loss: 1.3476 - val_accuracy: 0.4727
Epoch 19/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.3454 - accuracy:
0.4760
Epoch 19: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.3454 -
accuracy: 0.4760 - val_loss: 1.3485 - val_accuracy: 0.4738
Epoch 20/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.3427 - accuracy:
0.4779
Epoch 20: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 50ms/step - loss: 1.3427 -
accuracy: 0.4779 - val_loss: 1.3392 - val_accuracy: 0.4773
Epoch 21/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.3402 - accuracy:
0.4783
Epoch 21: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 49ms/step - loss: 1.3402 -
accuracy: 0.4783 - val_loss: 1.3451 - val_accuracy: 0.4750
Epoch 22/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.3377 - accuracy:
0.4794
Epoch 22: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 49ms/step - loss: 1.3377 -
accuracy: 0.4795 - val_loss: 1.3414 - val_accuracy: 0.4759
Epoch 23/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.3349 - accuracy:
0.4805
Epoch 23: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.3349 -
accuracy: 0.4805 - val_loss: 1.3373 - val_accuracy: 0.4769
Epoch 24/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.3323 - accuracy:
0.4817
Epoch 24: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.3323 -
accuracy: 0.4817 - val_loss: 1.3413 - val_accuracy: 0.4763
Epoch 25/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.3309 - accuracy:
```

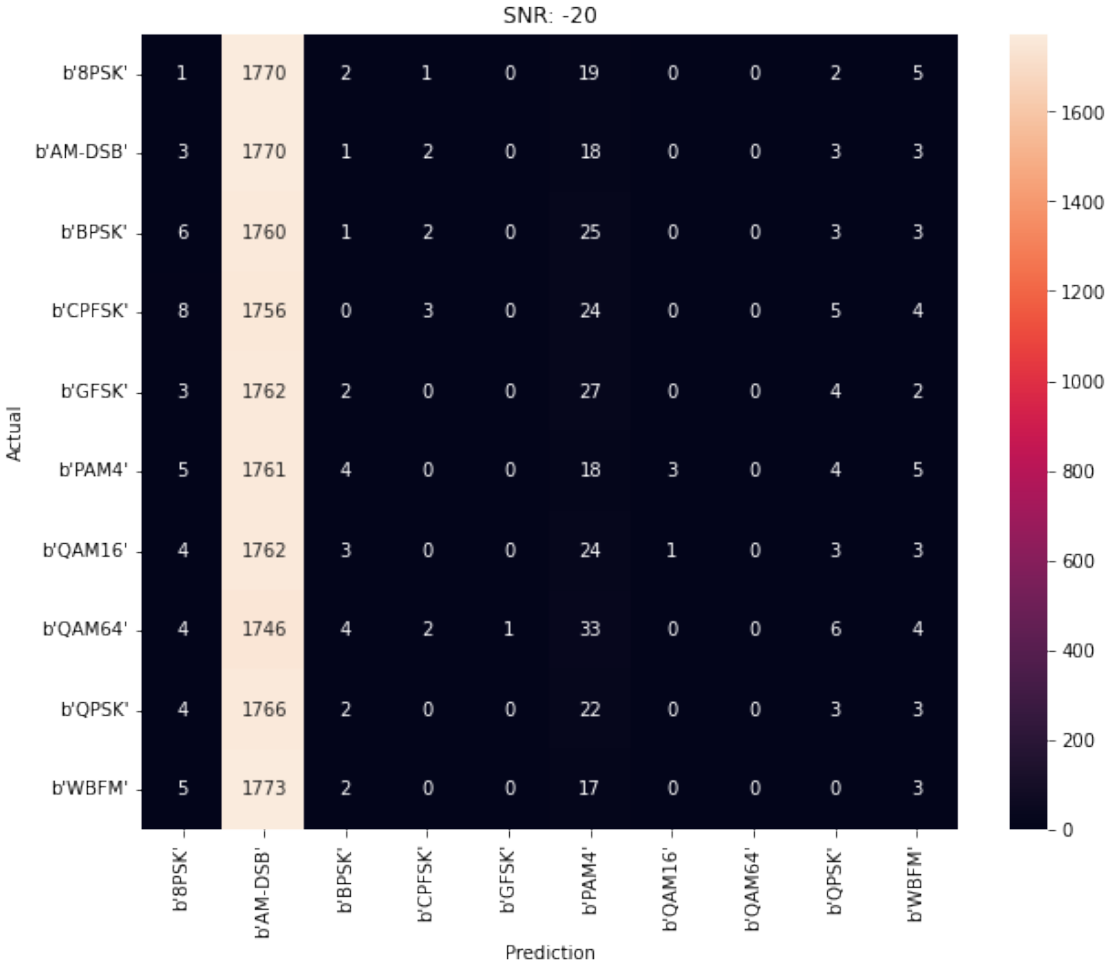
```
0.4824
Epoch 25: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 50ms/step - loss: 1.3309 -
accuracy: 0.4824 - val_loss: 1.3414 - val_accuracy: 0.4770
Epoch 26/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.3292 - accuracy:
0.4834
Epoch 26: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.3292 -
accuracy: 0.4834 - val_loss: 1.3465 - val_accuracy: 0.4760
Epoch 27/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.3272 - accuracy:
0.4841
Epoch 27: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 49ms/step - loss: 1.3272 -
accuracy: 0.4841 - val_loss: 1.3399 - val_accuracy: 0.4764
Epoch 28/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.3252 - accuracy:
0.4853
Epoch 28: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 49ms/step - loss: 1.3252 -
accuracy: 0.4853 - val_loss: 1.3387 - val_accuracy: 0.4783
```

[ ]:
```
plot_model_history(history, 'CNN Model')
model_scoring(cnn_lstm_model_2, X_test, testing_pair_labels)
```



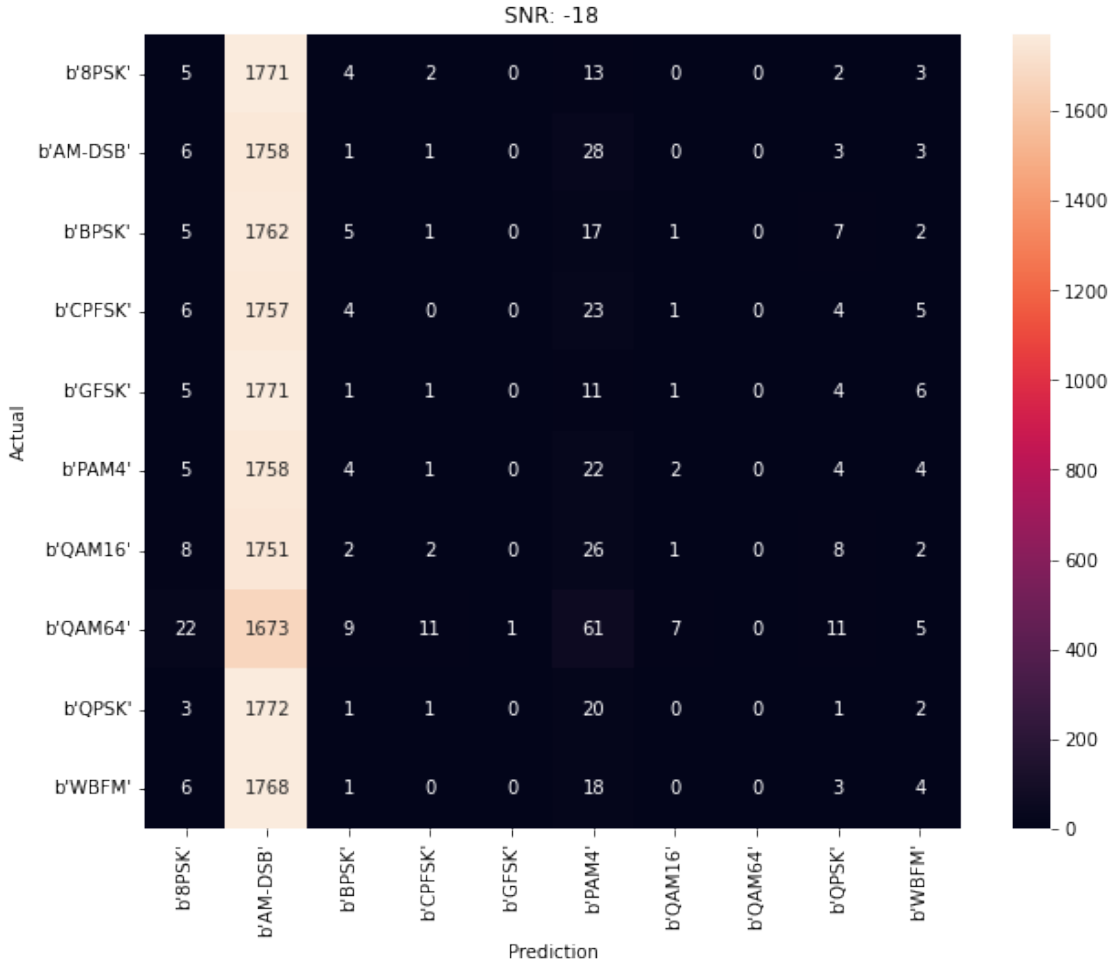
```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "
```

```
Accuracy at SNR = -20 is 0.1%
```

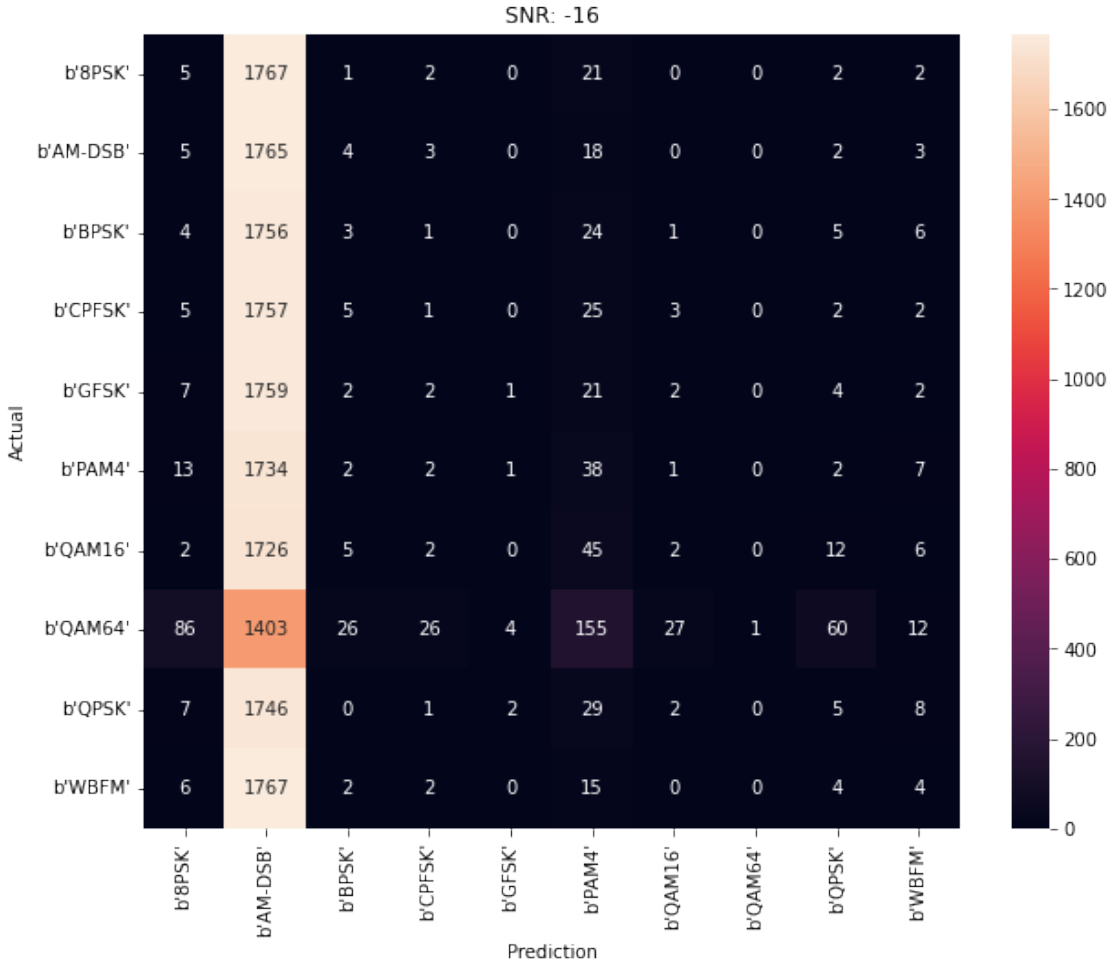



```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "
```

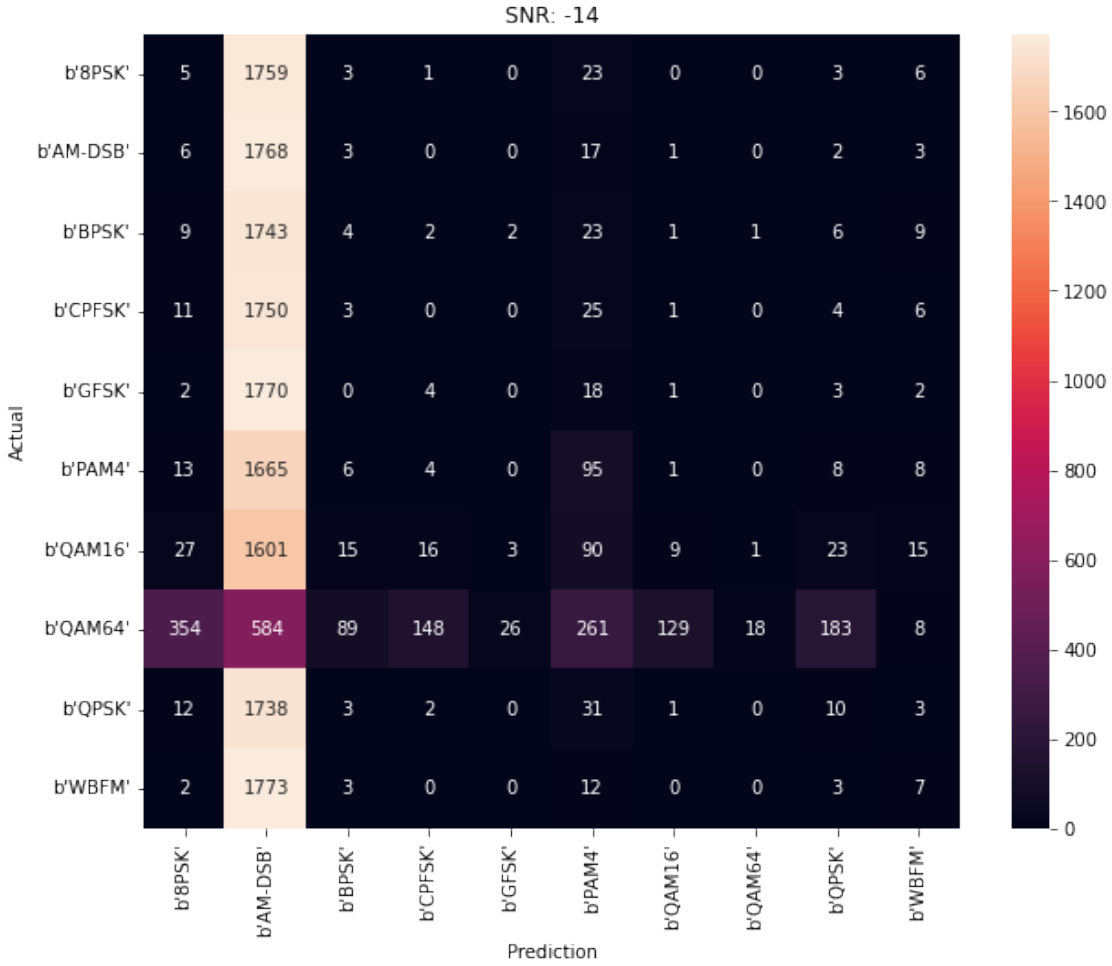
```
Accuracy at SNR = -18 is 0.09977777777777778%
```



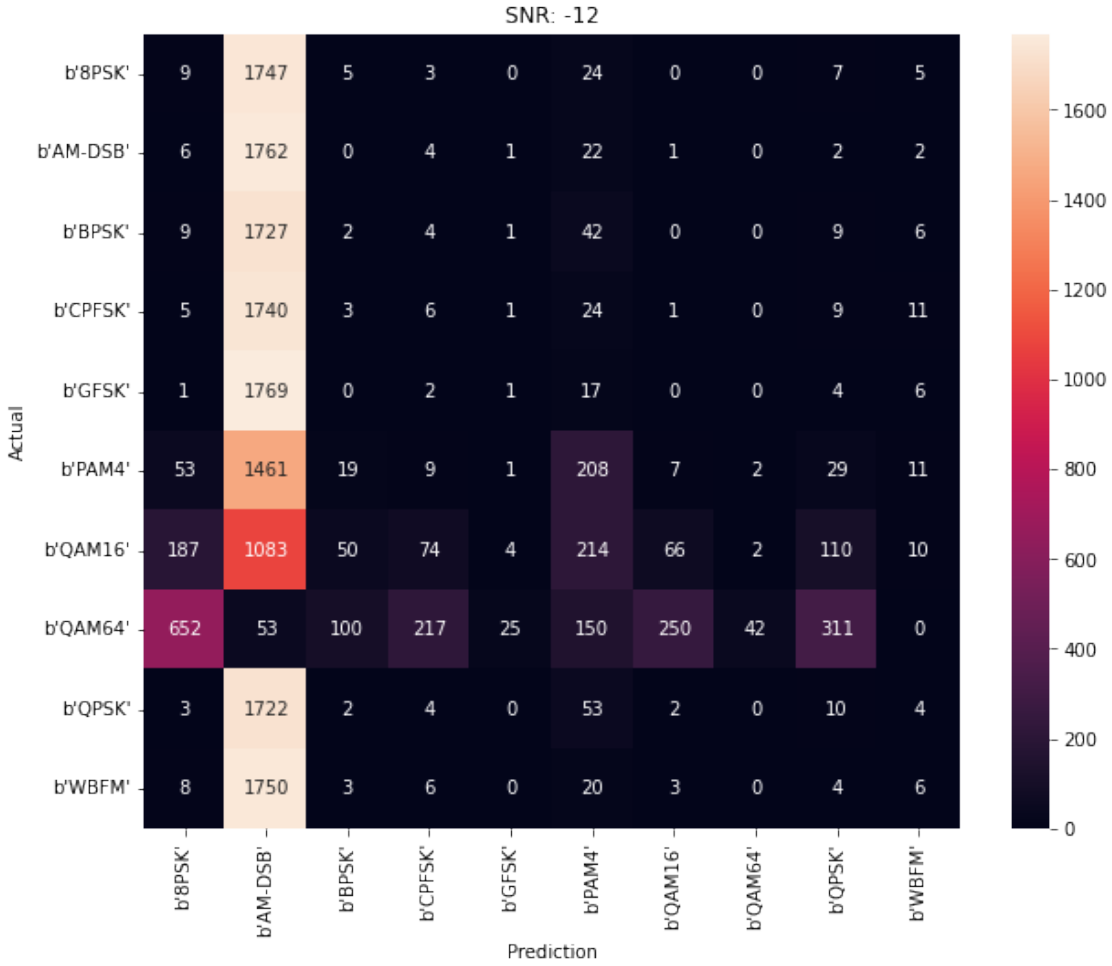

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = -16 is 0.10138888888888889%
```



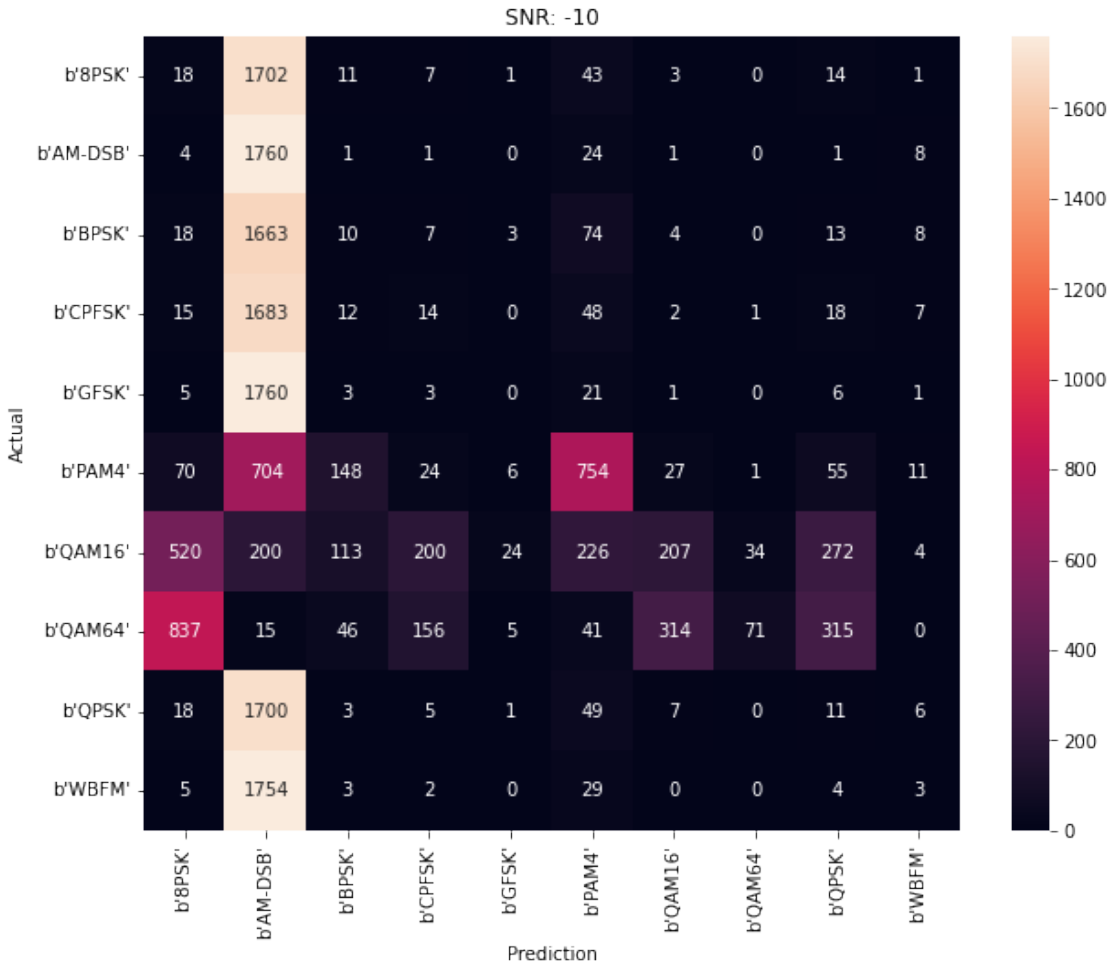

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = -14 is 0.10644444444444444%
```



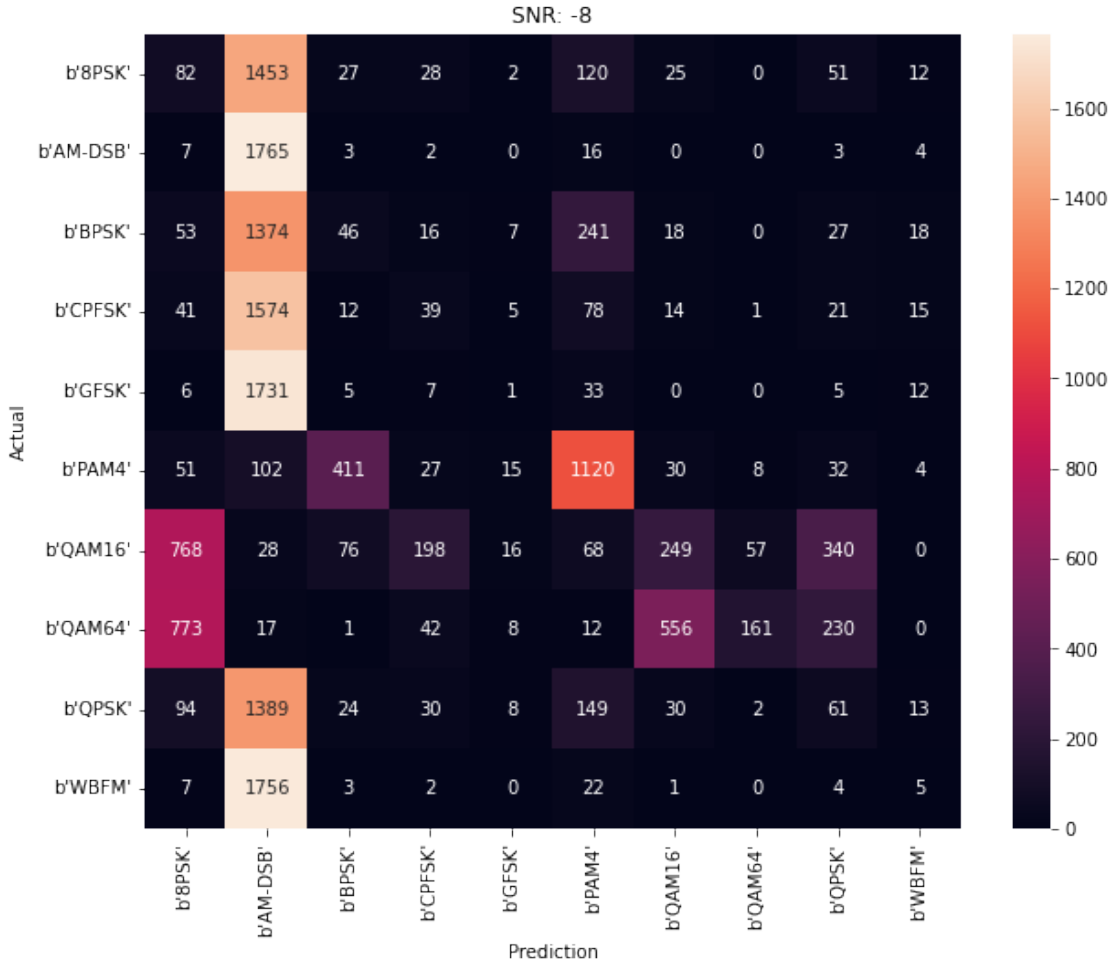

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = -12 is 0.11733333333333333%
```



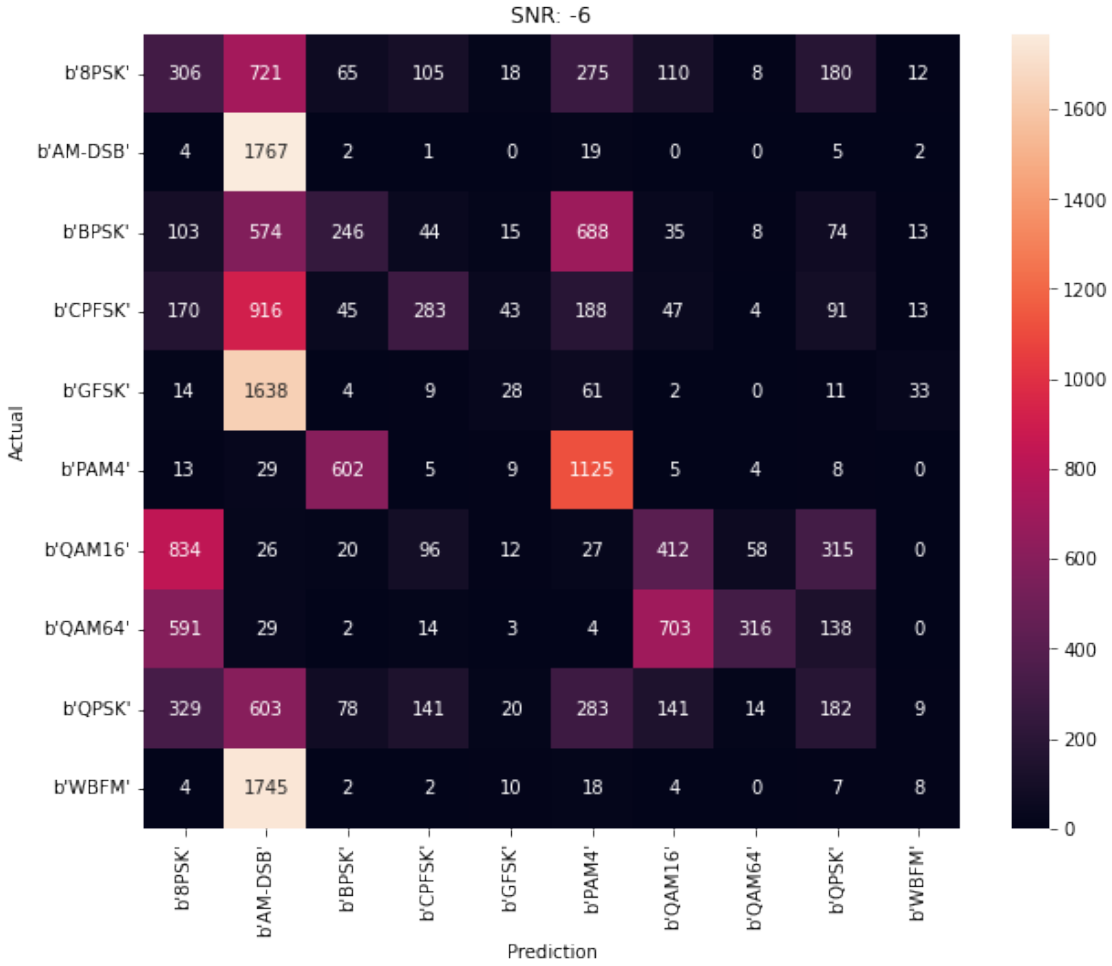

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = -10 is 0.1582222222222222%
```



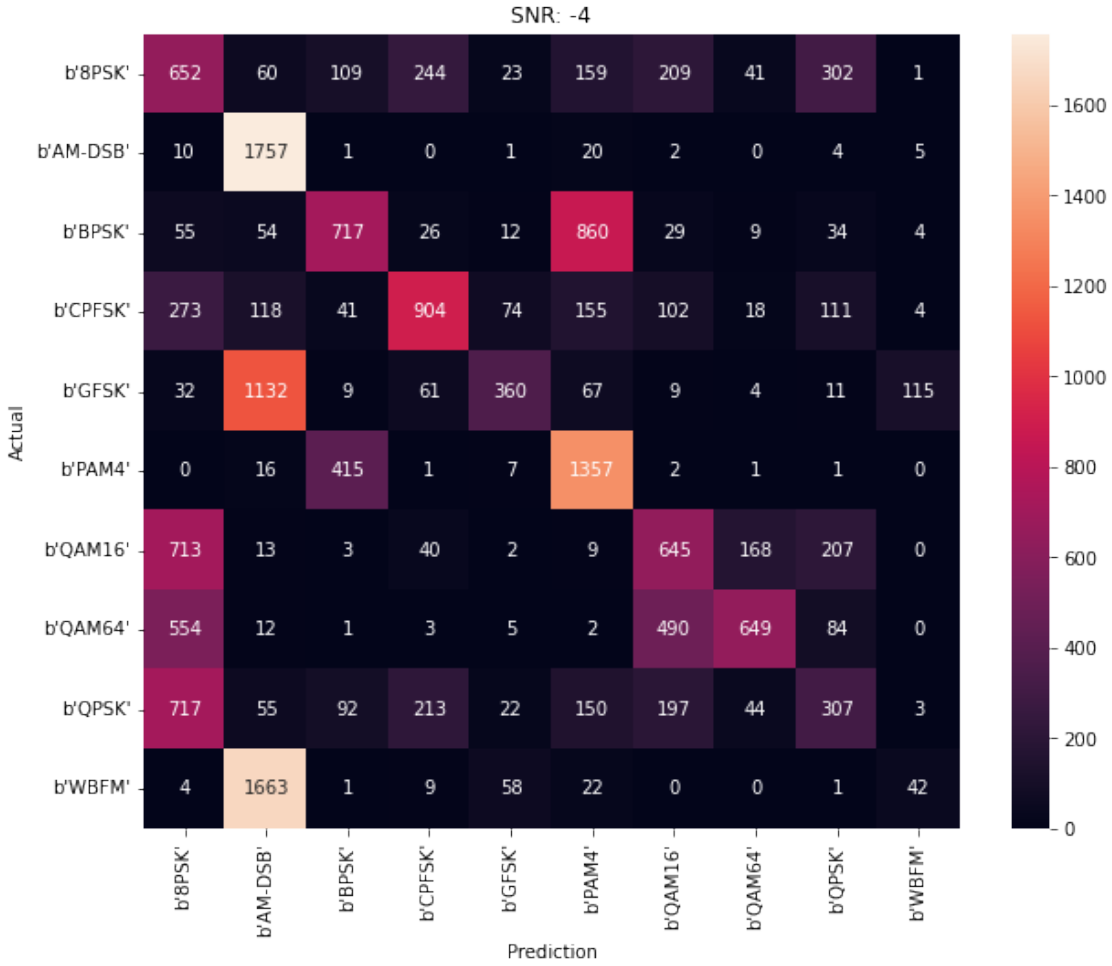

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = -8 is 0.19605555555555557%
```



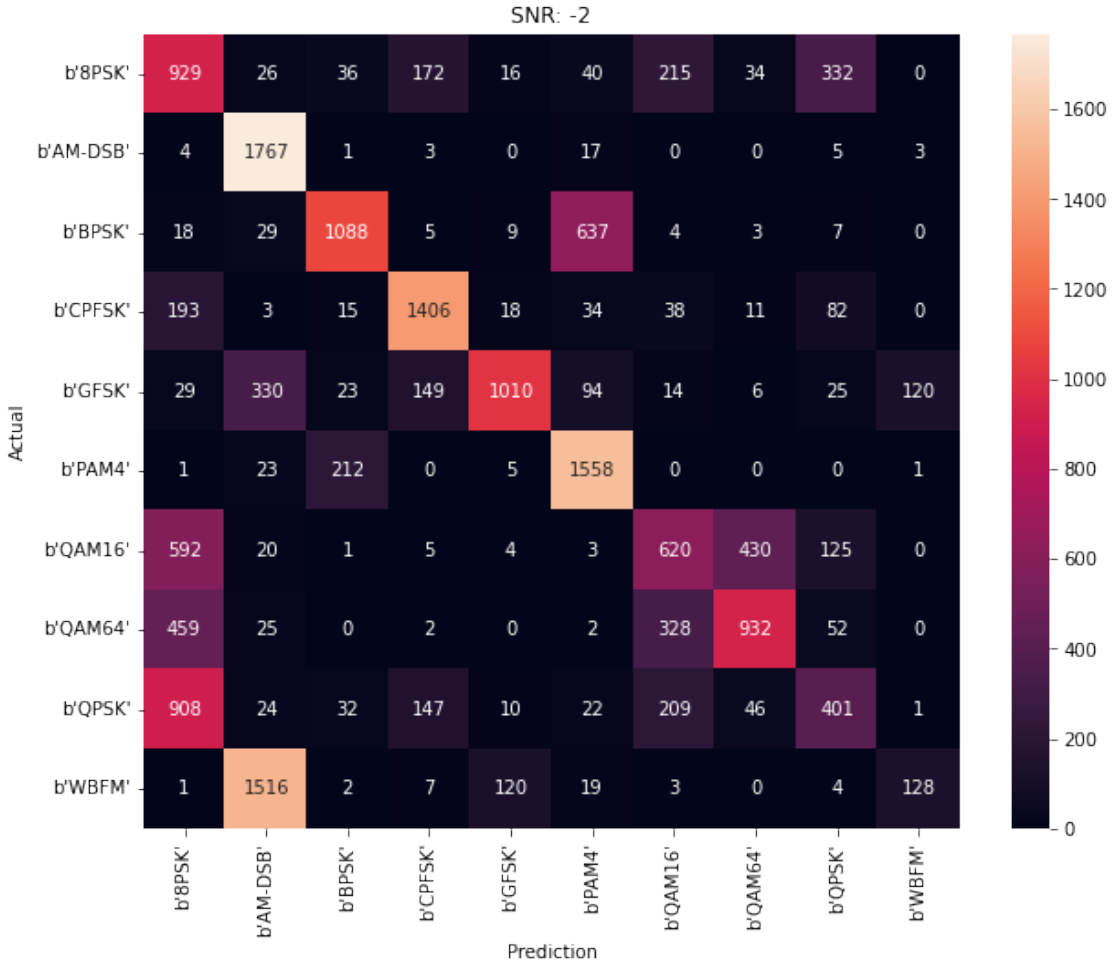

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = -6 is 0.2596111111111111%
```



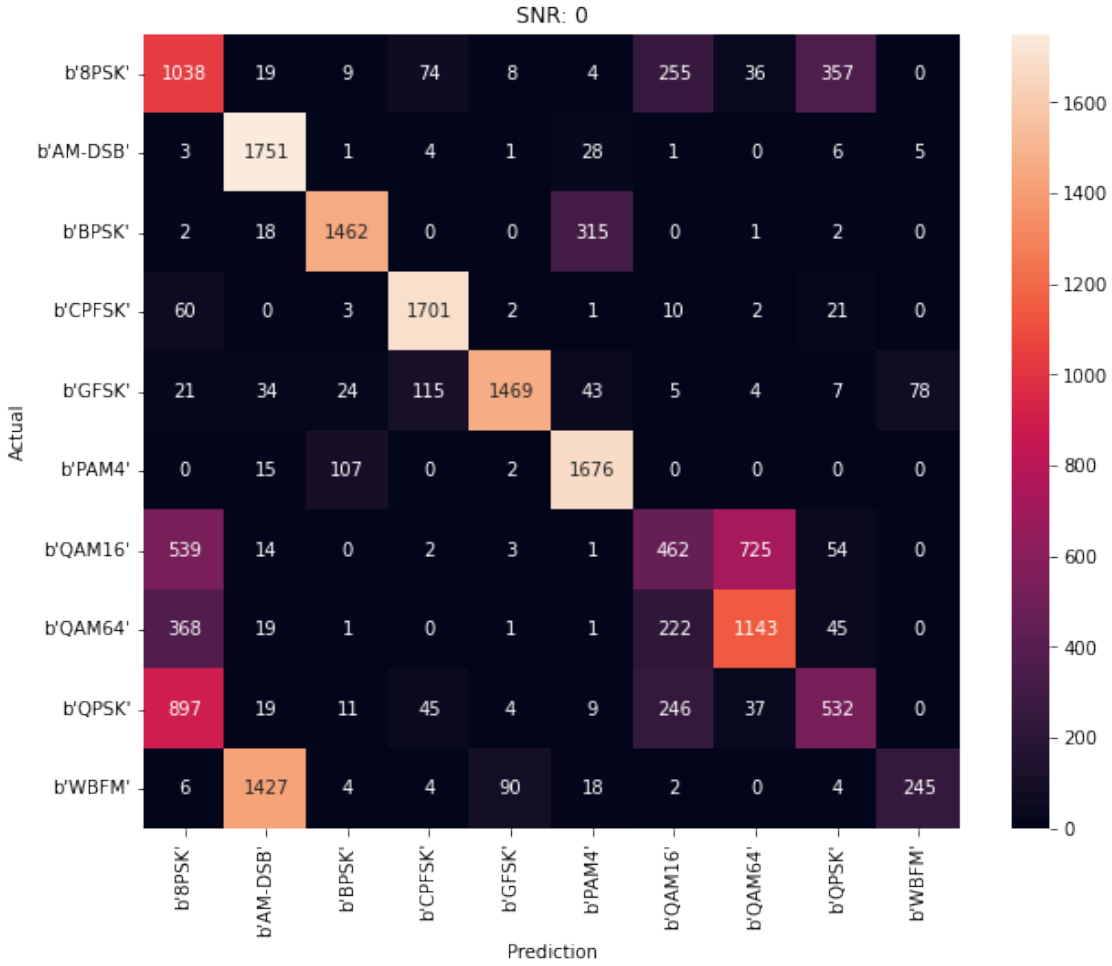

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = -4 is 0.41055555555555556%
```



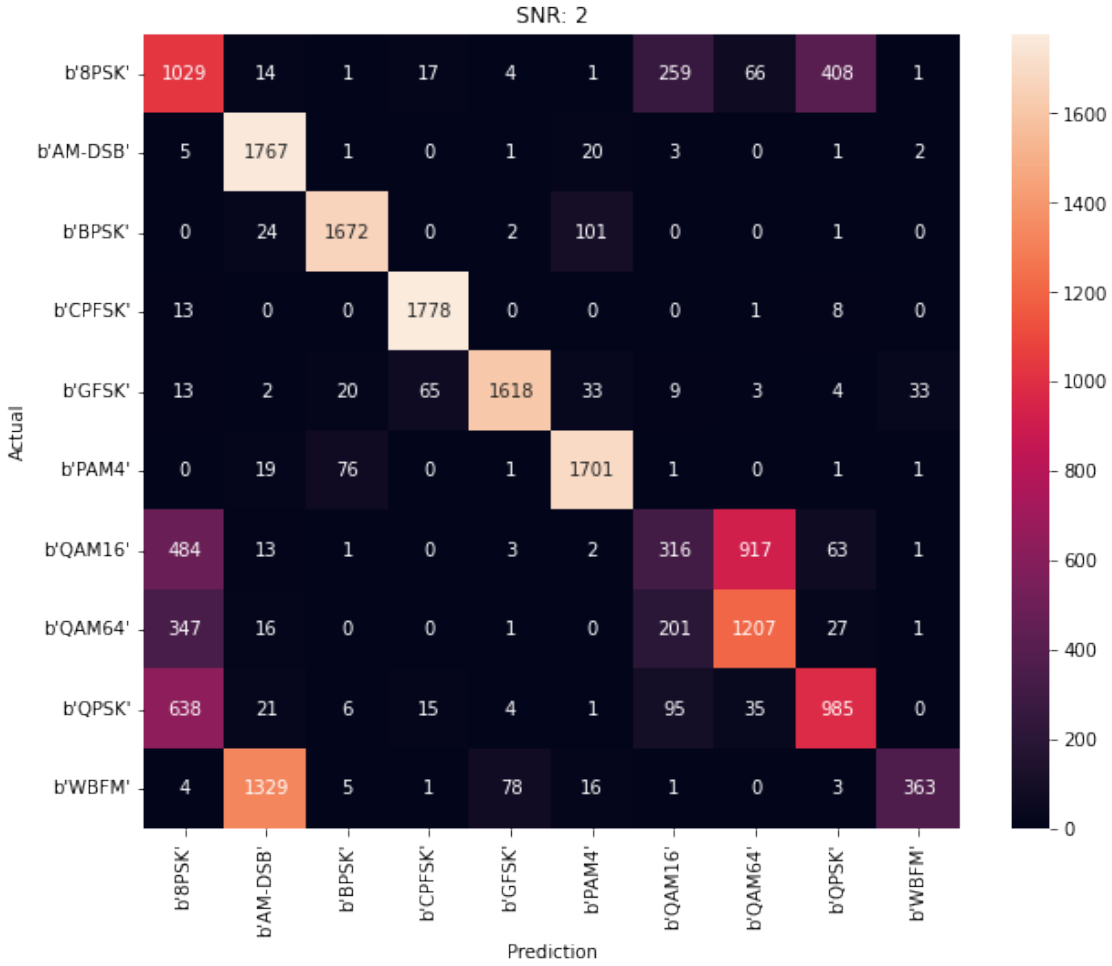

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = -2 is 0.5466111111111112%
```



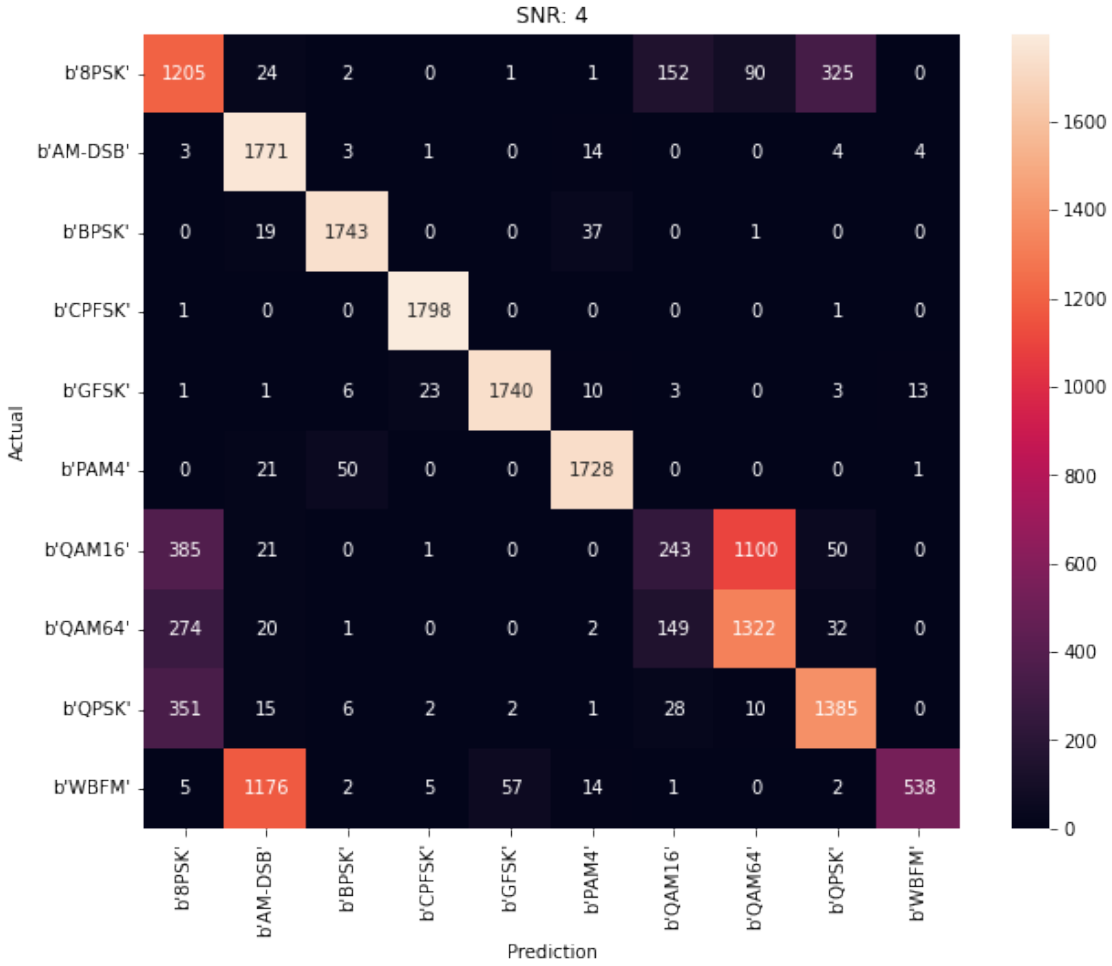

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = 0 is 0.6377222222222222%
```



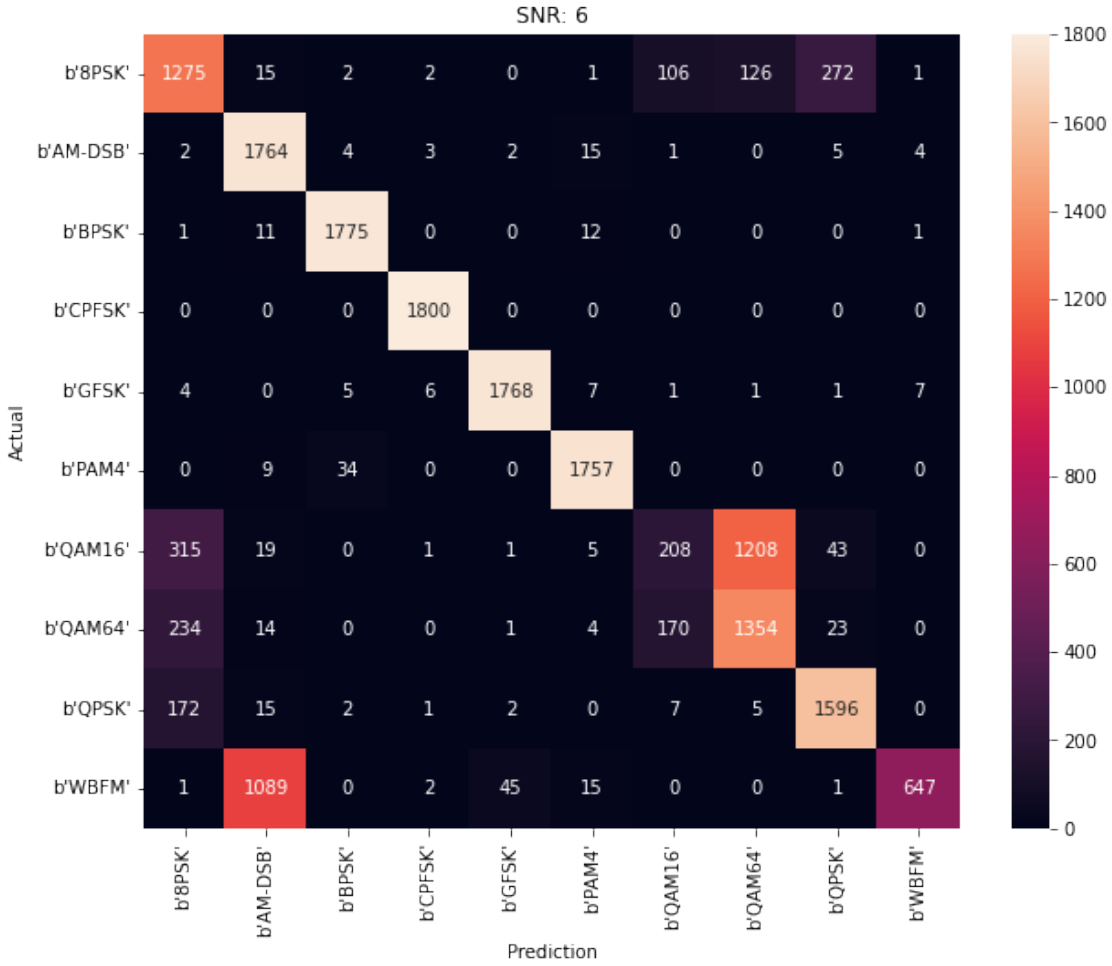

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = 2 is 0.6908888888888889%
```



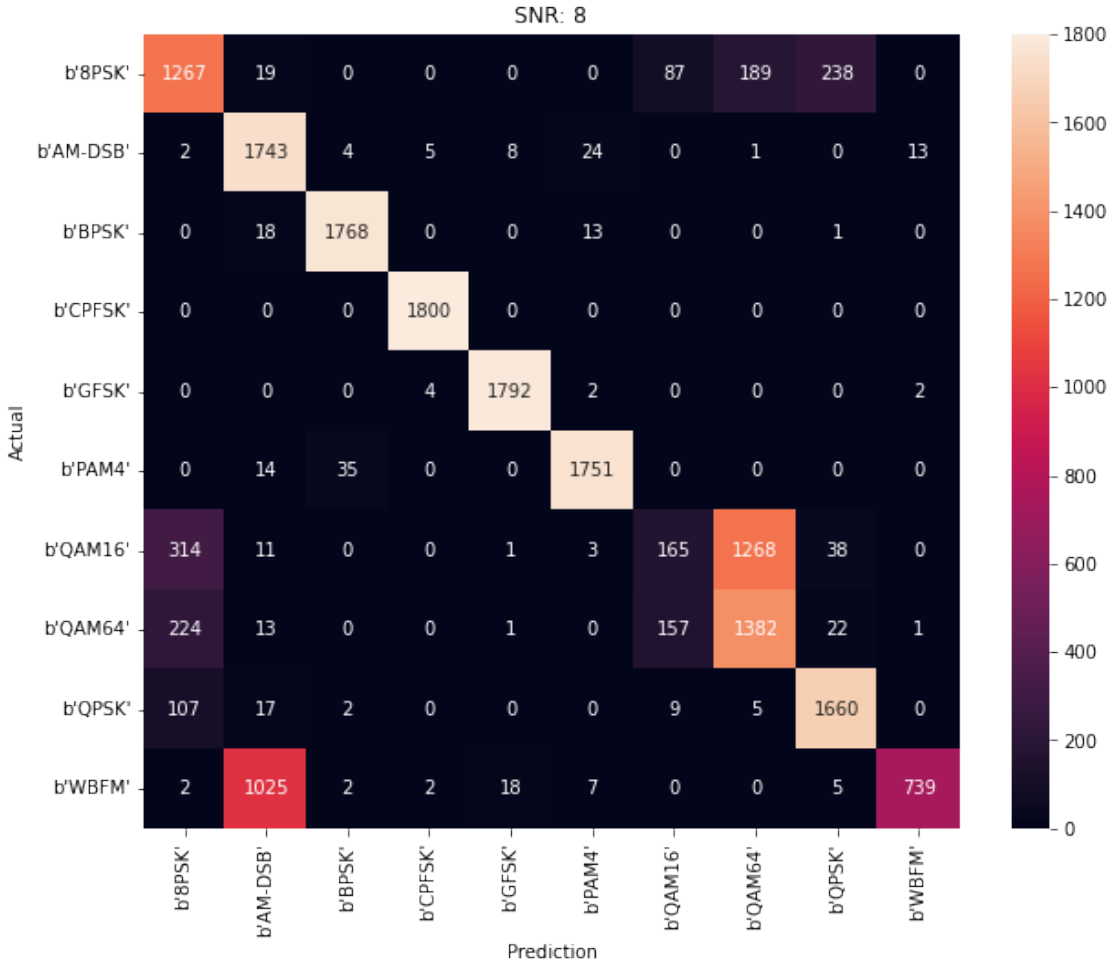

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = 4 is 0.7485%
```



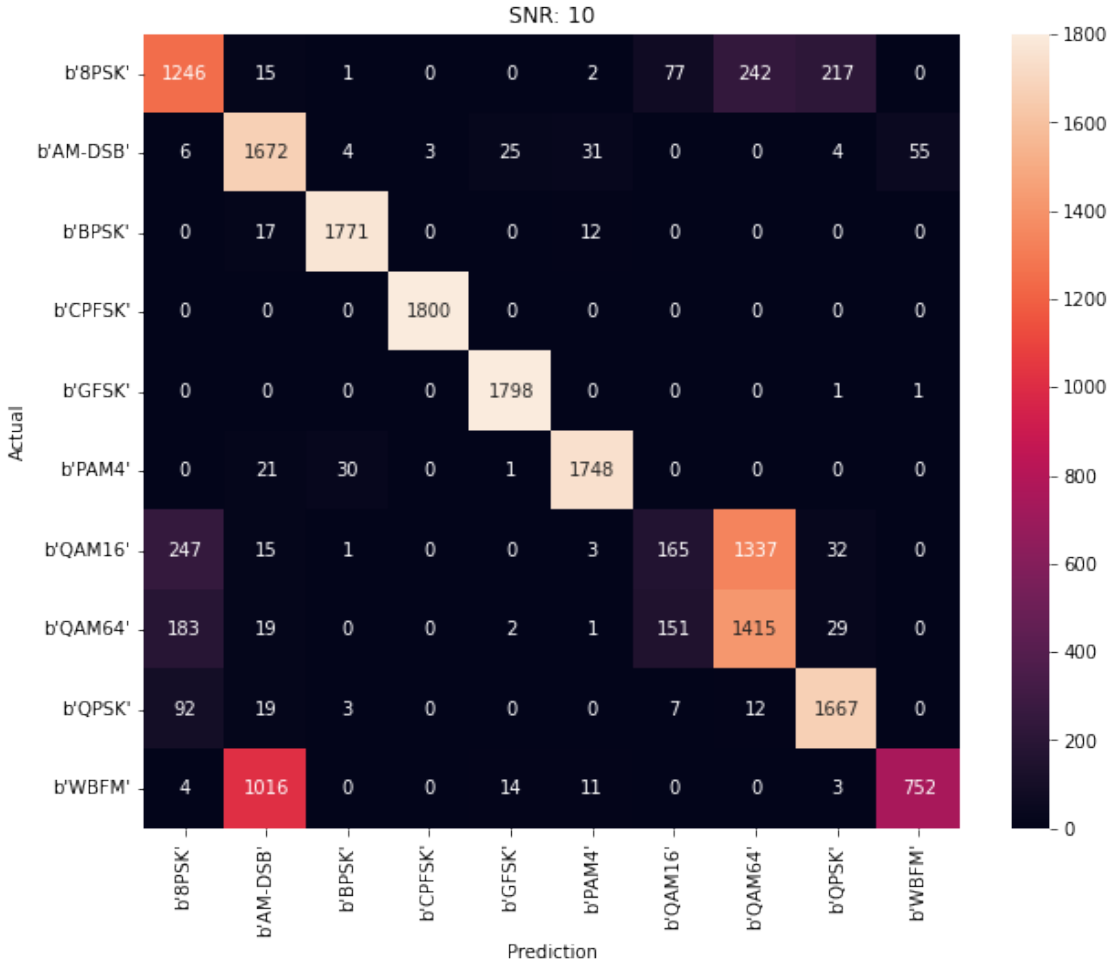

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = 6 is 0.7746666666666666%
```



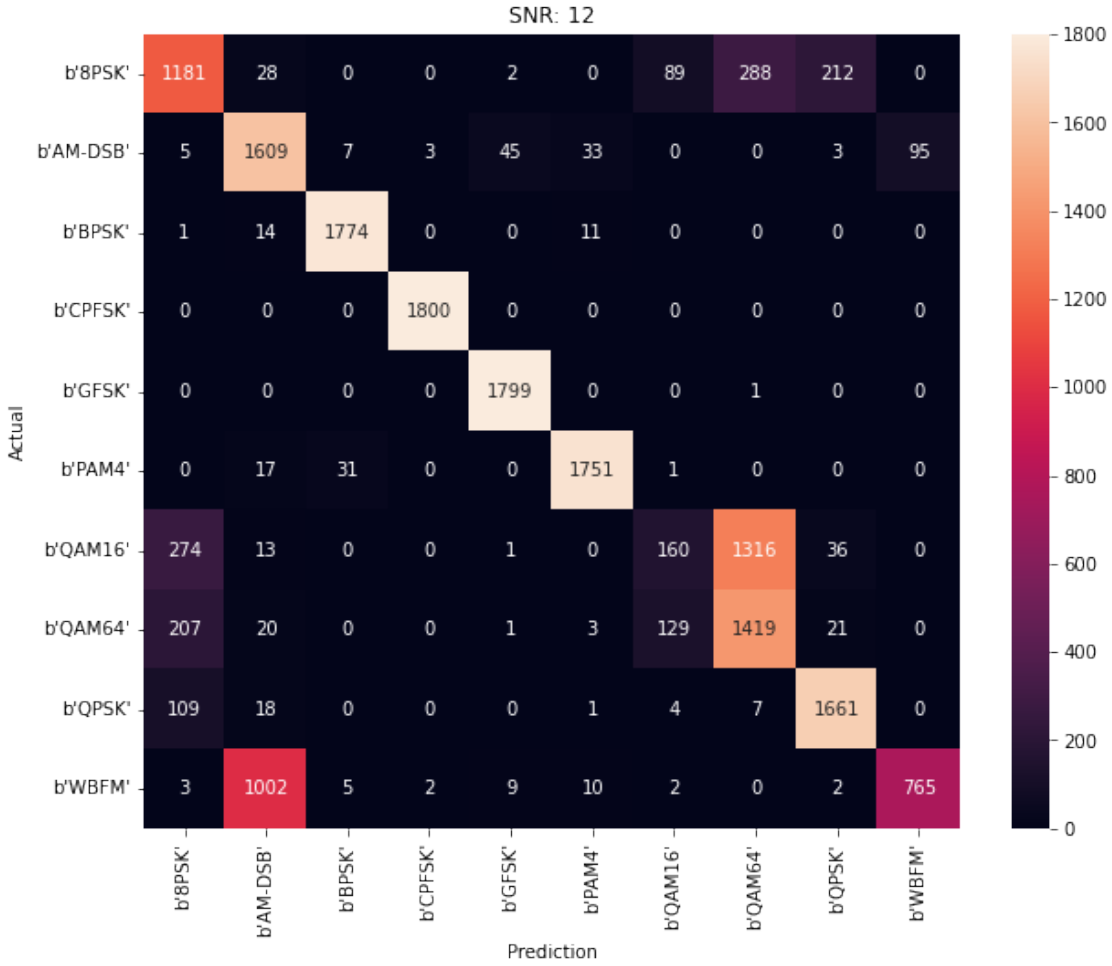

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = 8 is 0.7815%
```



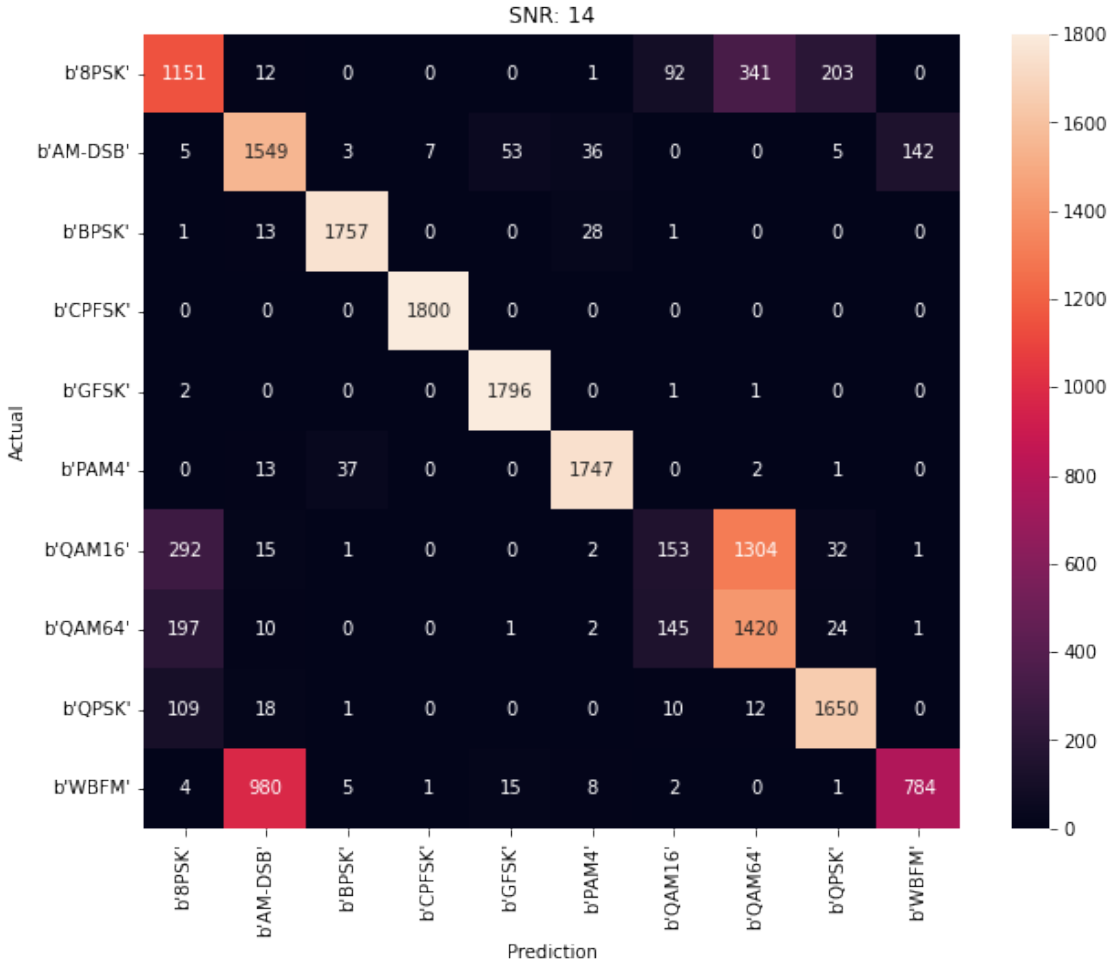

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = 10 is 0.7796666666666666%
```



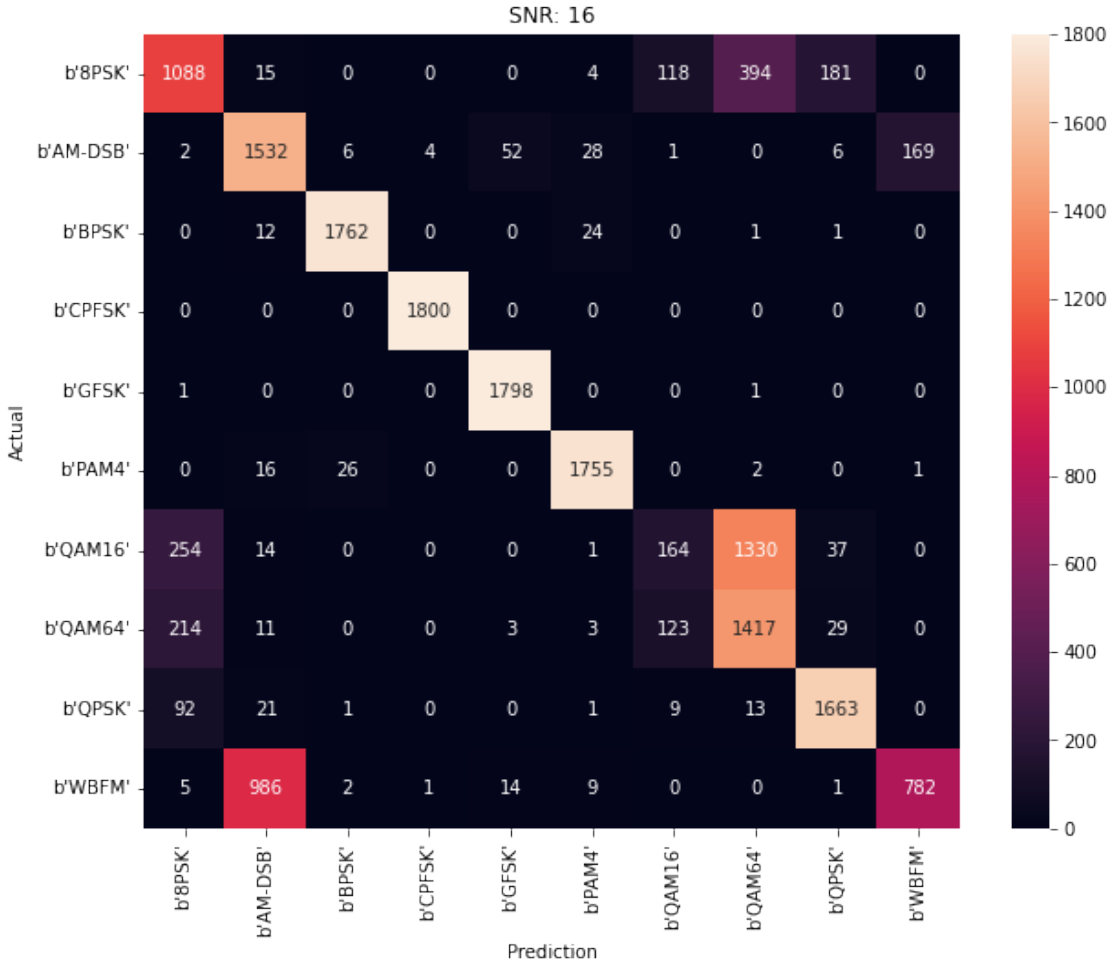

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = 12 is 0.7732777777777777%
```



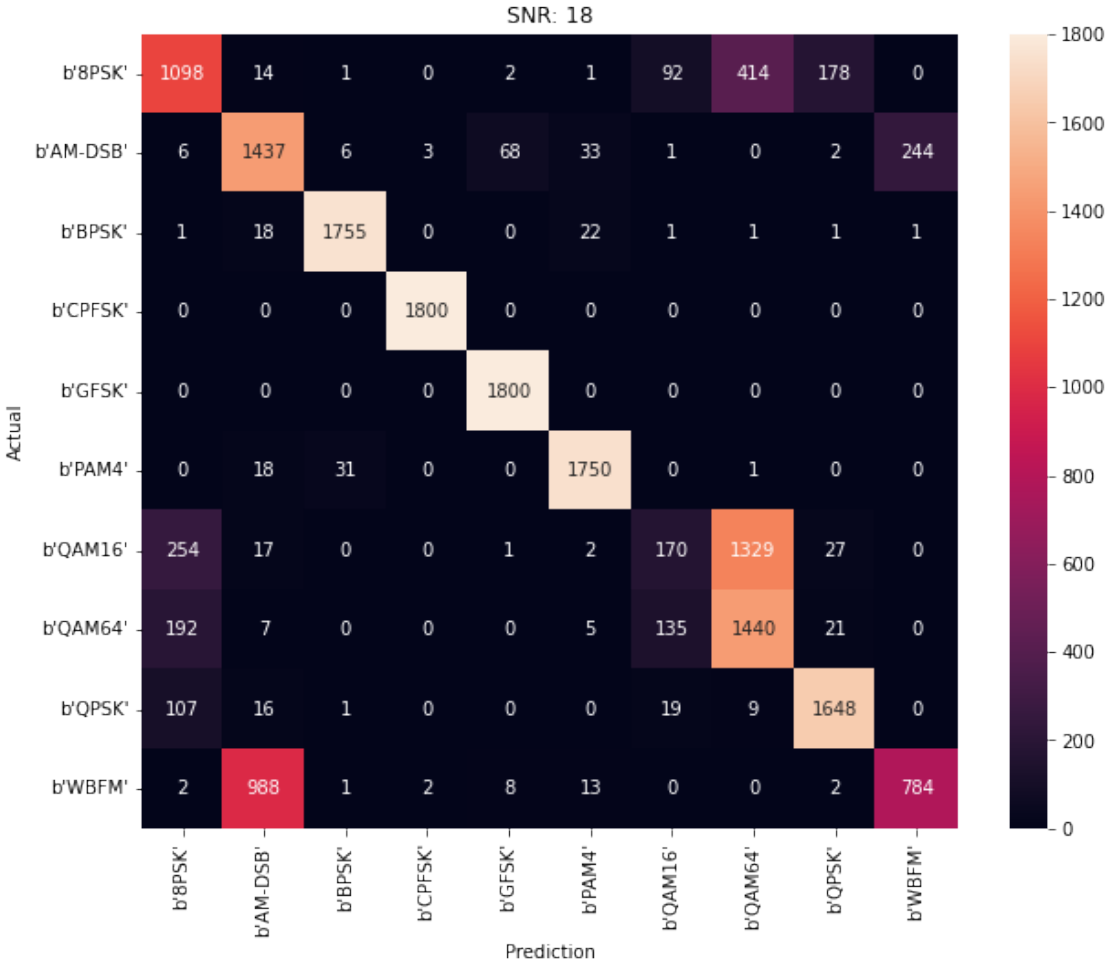

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = 14 is 0.7670555555555556%
```

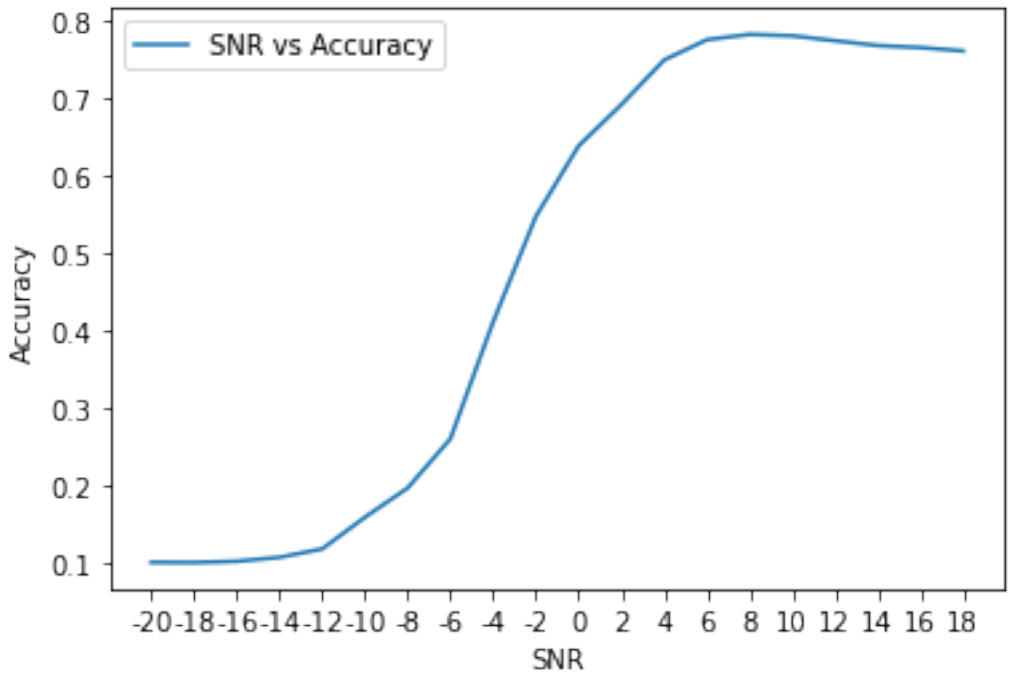



```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = 16 is 0.7645%
```




```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = 18 is 0.7601111111111111%
```


SNR: 18
b'8PSK' 1098 14 1 0 2 1 92 414 178 0
b'AM-DSB' 6 1437 6 3 68 33 1 0 2 244
b'BPSK' 1 18 1755 0 0 22 1 1 1 1
b'CPFSK' 0 0 0 1800 0 0 0 0 0 0
b'GFSK' 0 0 0 0 1800 0 0 0 0 0
b'PAM4' 0 18 31 0 0 1750 0 1 0 0
b'QAM16' 254 17 0 0 1 2 170 1329 27 0
b'QAM64' 192 7 0 0 0 5 135 1440 21 0
b'QPSK' 107 16 1 0 0 0 19 9 1648 0
b'WBFM' 2 988 1 2 8 13 0 0 2 784
Actual
b'8PSK' b'AM-DSB' b'BPSK' b'CPFSK' b'GFSK' b'PAM4' b'QAM16' b'QAM64' b'QPSK' b'WBFM'
Prediction
1800 1600 1400 1200 1000 800 600 400 200 0



7 Integrated Features Space

[18]:
```
fiit_training_data = integrate.cumtrapz(training_data, initial=0)
fiit_validation_data = integrate.cumtrapz(validation_data, initial=0)
fiit_testing_data = integrate.cumtrapz(testing_data, initial=0)
```

[ ]:
```
print('fiit training data shape:', fiit_training_data.shape)
print('fiit validation data shape:', fiit_validation_data.shape)
print('fiit testing data shape:', fiit_testing_data.shape)
```

```
fiit training data shape: (798000, 2, 128)
fiit validation data shape: (42000, 2, 128)
fiit testing data shape: (360000, 2, 128)
```

[ ]:
```
X_trainp = np.asarray(np.transpose(fiit_training_data, axes=(0,2,1)))
X_valp   = np.asarray(np.transpose(fiit_validation_data  , axes=(0,2,1)))
n_timesteps, n_features, n_outputs = X_trainp.shape[1], X_trainp.shape[2],␣
 ↪validation_onehot.shape[1]
n_steps, n_length = 4, 32
X_trainp = X_trainp.reshape((X_trainp.shape[0], n_steps, n_length, n_features))
X_valp = X_valp.reshape((X_valp.shape[0], n_steps, n_length, n_features))
```

[ ]:
```
X_test = np.asarray(np.transpose(fiit_testing_data, axes=(0,2,1)))
n_timesteps, n_features, n_outputs = X_test.shape[1], X_test.shape[2],␣
 ↪validation_onehot.shape[1]
n_steps, n_length = 4, 32
X_test = X_test.reshape((X_test.shape[0], n_steps, n_length, n_features))
```

[ ]:
```
with tf.device('/device:GPU:0'):
  history = cnn_lstm_model_2.fit(X_trainp, training_onehot, batch_size=512,␣
 ↪epochs=epochs, validation_data=(X_valp, validation_onehot), callbacks=[es,␣
 ↪checkpointer], verbose=1)
```

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Epoch 1/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.7231 - accuracy:
0.3158
Epoch 1: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 50ms/step - loss: 1.7230 -
accuracy: 0.3158 - val_loss: 1.5369 - val_accuracy: 0.3700
Epoch 2/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.5094 - accuracy:
0.3792
Epoch 2: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 49ms/step - loss: 1.5094 -
accuracy: 0.3792 - val_loss: 1.4394 - val_accuracy: 0.4029
Epoch 3/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.4548 - accuracy:
0.3997
Epoch 3: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.4548 -
accuracy: 0.3997 - val_loss: 1.4220 - val_accuracy: 0.4115
Epoch 4/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.4206 - accuracy:
0.4126
Epoch 4: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 49ms/step - loss: 1.4206 -
accuracy: 0.4125 - val_loss: 1.3971 - val_accuracy: 0.4253
Epoch 5/200
1559/1559 [==============================] - ETA: 0s - loss: 1.3945 - accuracy:
0.4231
Epoch 5: val_loss did not improve from 0.97569
1559/1559 [==============================] - 76s 48ms/step - loss: 1.3945 -
accuracy: 0.4231 - val_loss: 1.3495 - val_accuracy: 0.4398
```

```
Epoch 6/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.3748 - accuracy:
0.4311
Epoch 6: val_loss did not improve from 0.97569
1559/1559 [==============================] - 79s 51ms/step - loss: 1.3748 -
accuracy: 0.4311 - val_loss: 1.3432 - val_accuracy: 0.4412
Epoch 7/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.3599 - accuracy:
0.4365
Epoch 7: val_loss did not improve from 0.97569
1559/1559 [==============================] - 76s 49ms/step - loss: 1.3599 -
accuracy: 0.4365 - val_loss: 1.3437 - val_accuracy: 0.4440
Epoch 8/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.3478 - accuracy:
0.4414
Epoch 8: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 49ms/step - loss: 1.3479 -
accuracy: 0.4414 - val_loss: 1.3325 - val_accuracy: 0.4491
Epoch 9/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.3369 - accuracy:
0.4459
Epoch 9: val_loss did not improve from 0.97569
1559/1559 [==============================] - 76s 49ms/step - loss: 1.3369 -
accuracy: 0.4459 - val_loss: 1.3194 - val_accuracy: 0.4554
Epoch 10/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.3277 - accuracy:
0.4492
Epoch 10: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 50ms/step - loss: 1.3277 -
accuracy: 0.4492 - val_loss: 1.3005 - val_accuracy: 0.4586
Epoch 11/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.3189 - accuracy:
0.4522
Epoch 11: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 50ms/step - loss: 1.3189 -
accuracy: 0.4522 - val_loss: 1.2916 - val_accuracy: 0.4633
Epoch 12/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.3120 - accuracy:
0.4553
Epoch 12: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.3121 -
accuracy: 0.4553 - val_loss: 1.3044 - val_accuracy: 0.4596
Epoch 13/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.3044 - accuracy:
0.4587
Epoch 13: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.3044 -
accuracy: 0.4587 - val_loss: 1.2828 - val_accuracy: 0.4658
```

```
Epoch 14/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.2981 - accuracy:
0.4610
Epoch 14: val_loss did not improve from 0.97569
1559/1559 [==============================] - 79s 51ms/step - loss: 1.2981 -
accuracy: 0.4611 - val_loss: 1.2982 - val_accuracy: 0.4615
Epoch 15/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.2919 - accuracy:
0.4638
Epoch 15: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.2920 -
accuracy: 0.4637 - val_loss: 1.2782 - val_accuracy: 0.4649
Epoch 16/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.2863 - accuracy:
0.4659
Epoch 16: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 50ms/step - loss: 1.2863 -
accuracy: 0.4659 - val_loss: 1.2655 - val_accuracy: 0.4760
Epoch 17/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.2805 - accuracy:
0.4685
Epoch 17: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 50ms/step - loss: 1.2805 -
accuracy: 0.4685 - val_loss: 1.2696 - val_accuracy: 0.4728
Epoch 18/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.2734 - accuracy:
0.4703
Epoch 18: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 49ms/step - loss: 1.2734 -
accuracy: 0.4703 - val_loss: 1.2843 - val_accuracy: 0.4678
Epoch 19/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.2682 - accuracy:
0.4730
Epoch 19: val_loss did not improve from 0.97569
1559/1559 [==============================] - 79s 50ms/step - loss: 1.2682 -
accuracy: 0.4730 - val_loss: 1.2479 - val_accuracy: 0.4792
Epoch 20/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.2618 - accuracy:
0.4755
Epoch 20: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.2618 -
accuracy: 0.4755 - val_loss: 1.2385 - val_accuracy: 0.4844
Epoch 21/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.2559 - accuracy:
0.4776
Epoch 21: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.2559 -
accuracy: 0.4776 - val_loss: 1.2539 - val_accuracy: 0.4794
```

```
Epoch 22/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.2493 - accuracy:
0.4796
Epoch 22: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 50ms/step - loss: 1.2493 -
accuracy: 0.4796 - val_loss: 1.2370 - val_accuracy: 0.4846
Epoch 23/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.2419 - accuracy:
0.4830
Epoch 23: val_loss did not improve from 0.97569
1559/1559 [==============================] - 79s 50ms/step - loss: 1.2419 -
accuracy: 0.4830 - val_loss: 1.2325 - val_accuracy: 0.4855
Epoch 24/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.2352 - accuracy:
0.4859
Epoch 24: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 50ms/step - loss: 1.2352 -
accuracy: 0.4859 - val_loss: 1.2162 - val_accuracy: 0.4922
Epoch 25/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.2283 - accuracy:
0.4878
Epoch 25: val_loss did not improve from 0.97569
1559/1559 [==============================] - 79s 51ms/step - loss: 1.2283 -
accuracy: 0.4878 - val_loss: 1.2484 - val_accuracy: 0.4809
Epoch 26/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.2202 - accuracy:
0.4909
Epoch 26: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 49ms/step - loss: 1.2202 -
accuracy: 0.4909 - val_loss: 1.2243 - val_accuracy: 0.4894
Epoch 27/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.2118 - accuracy:
0.4937
Epoch 27: val_loss did not improve from 0.97569
1559/1559 [==============================] - 76s 49ms/step - loss: 1.2118 -
accuracy: 0.4937 - val_loss: 1.2211 - val_accuracy: 0.4936
Epoch 28/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.2045 - accuracy:
0.4971
Epoch 28: val_loss did not improve from 0.97569
1559/1559 [==============================] - 79s 50ms/step - loss: 1.2045 -
accuracy: 0.4971 - val_loss: 1.2097 - val_accuracy: 0.4953
Epoch 29/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.1953 - accuracy:
0.4997
Epoch 29: val_loss did not improve from 0.97569
1559/1559 [==============================] - 76s 49ms/step - loss: 1.1953 -
accuracy: 0.4997 - val_loss: 1.1944 - val_accuracy: 0.4989
```

```
Epoch 30/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.1871 - accuracy:
0.5032
Epoch 30: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.1871 -
accuracy: 0.5032 - val_loss: 1.1832 - val_accuracy: 0.5019
Epoch 31/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.1781 - accuracy:
0.5061
Epoch 31: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 49ms/step - loss: 1.1782 -
accuracy: 0.5060 - val_loss: 1.1623 - val_accuracy: 0.5129
Epoch 32/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.1690 - accuracy:
0.5097
Epoch 32: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 50ms/step - loss: 1.1690 -
accuracy: 0.5096 - val_loss: 1.1523 - val_accuracy: 0.5148
Epoch 33/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.1606 - accuracy:
0.5124
Epoch 33: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 49ms/step - loss: 1.1606 -
accuracy: 0.5124 - val_loss: 1.1597 - val_accuracy: 0.5118
Epoch 34/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.1527 - accuracy:
0.5147
Epoch 34: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.1527 -
accuracy: 0.5147 - val_loss: 1.1744 - val_accuracy: 0.5097
Epoch 35/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.1438 - accuracy:
0.5181
Epoch 35: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.1438 -
accuracy: 0.5181 - val_loss: 1.1505 - val_accuracy: 0.5153
Epoch 36/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.1362 - accuracy:
0.5206
Epoch 36: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.1362 -
accuracy: 0.5206 - val_loss: 1.1279 - val_accuracy: 0.5242
Epoch 37/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.1283 - accuracy:
0.5233
Epoch 37: val_loss did not improve from 0.97569
1559/1559 [==============================] - 79s 51ms/step - loss: 1.1284 -
accuracy: 0.5233 - val_loss: 1.1585 - val_accuracy: 0.5134
```

```
Epoch 38/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.1205 - accuracy:
0.5256
Epoch 38: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.1205 -
accuracy: 0.5256 - val_loss: 1.1170 - val_accuracy: 0.5243
Epoch 39/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.1129 - accuracy:
0.5280
Epoch 39: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.1129 -
accuracy: 0.5280 - val_loss: 1.1113 - val_accuracy: 0.5285
Epoch 40/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.1065 - accuracy:
0.5302
Epoch 40: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.1065 -
accuracy: 0.5302 - val_loss: 1.1104 - val_accuracy: 0.5271
Epoch 41/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.1016 - accuracy:
0.5318
Epoch 41: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 50ms/step - loss: 1.1017 -
accuracy: 0.5318 - val_loss: 1.1013 - val_accuracy: 0.5278
Epoch 42/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0963 - accuracy:
0.5337
Epoch 42: val_loss did not improve from 0.97569
1559/1559 [==============================] - 79s 50ms/step - loss: 1.0963 -
accuracy: 0.5337 - val_loss: 1.0781 - val_accuracy: 0.5367
Epoch 43/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0917 - accuracy:
0.5346
Epoch 43: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 49ms/step - loss: 1.0917 -
accuracy: 0.5346 - val_loss: 1.0727 - val_accuracy: 0.5380
Epoch 44/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0870 - accuracy:
0.5369
Epoch 44: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.0870 -
accuracy: 0.5369 - val_loss: 1.0782 - val_accuracy: 0.5365
Epoch 45/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0835 - accuracy:
0.5383
Epoch 45: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.0835 -
accuracy: 0.5383 - val_loss: 1.0895 - val_accuracy: 0.5331
```

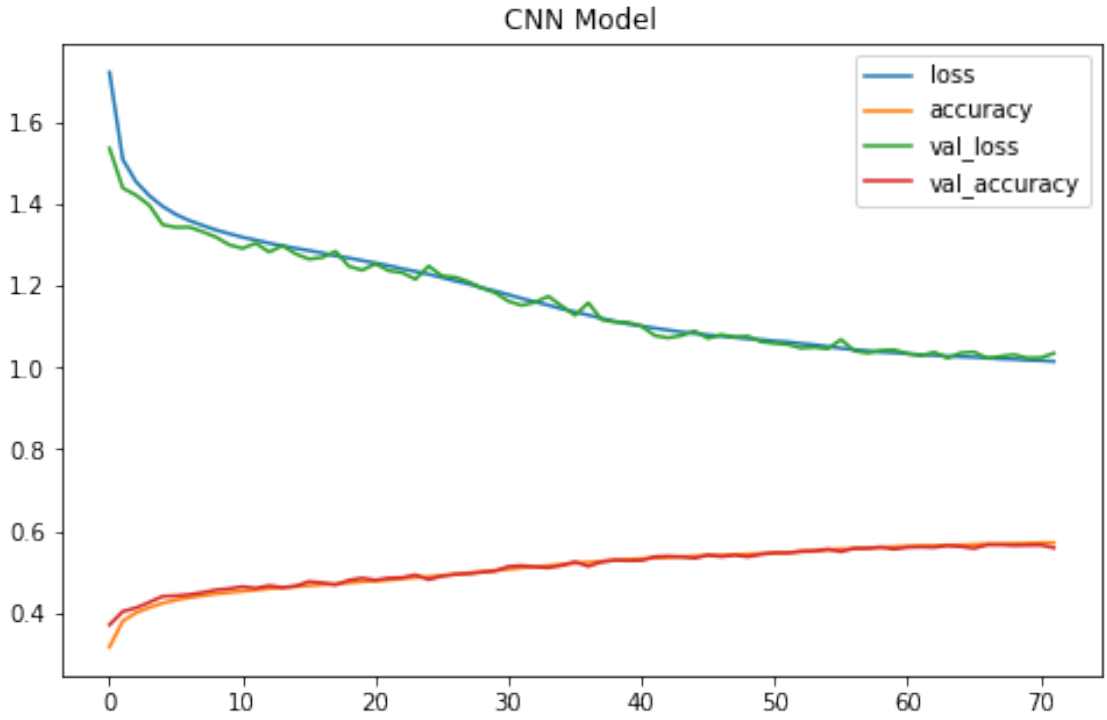
```
Epoch 46/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0797 - accuracy:
0.5396
Epoch 46: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.0797 -
accuracy: 0.5396 - val_loss: 1.0713 - val_accuracy: 0.5412
Epoch 47/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0768 - accuracy:
0.5407
Epoch 47: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.0768 -
accuracy: 0.5407 - val_loss: 1.0804 - val_accuracy: 0.5371
Epoch 48/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0737 - accuracy:
0.5419
Epoch 48: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.0737 -
accuracy: 0.5419 - val_loss: 1.0746 - val_accuracy: 0.5408
Epoch 49/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0702 - accuracy:
0.5430
Epoch 49: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.0702 -
accuracy: 0.5430 - val_loss: 1.0767 - val_accuracy: 0.5369
Epoch 50/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0684 - accuracy:
0.5444
Epoch 50: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 50ms/step - loss: 1.0684 -
accuracy: 0.5444 - val_loss: 1.0629 - val_accuracy: 0.5427
Epoch 51/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0647 - accuracy:
0.5462
Epoch 51: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 50ms/step - loss: 1.0648 -
accuracy: 0.5462 - val_loss: 1.0592 - val_accuracy: 0.5463
Epoch 52/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0627 - accuracy:
0.5472
Epoch 52: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.0627 -
accuracy: 0.5472 - val_loss: 1.0564 - val_accuracy: 0.5460
Epoch 53/200
1559/1559 [==============================] - ETA: 0s - loss: 1.0586 - accuracy:
0.5491
Epoch 53: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.0586 -
accuracy: 0.5491 - val_loss: 1.0475 - val_accuracy: 0.5502
```

```
Epoch 54/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0557 - accuracy:
0.5503
Epoch 54: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.0557 -
accuracy: 0.5503 - val_loss: 1.0495 - val_accuracy: 0.5510
Epoch 55/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0515 - accuracy:
0.5533
Epoch 55: val_loss did not improve from 0.97569
1559/1559 [==============================] - 79s 51ms/step - loss: 1.0515 -
accuracy: 0.5533 - val_loss: 1.0458 - val_accuracy: 0.5553
Epoch 56/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0472 - accuracy:
0.5554
Epoch 56: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.0472 -
accuracy: 0.5554 - val_loss: 1.0685 - val_accuracy: 0.5492
Epoch 57/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0437 - accuracy:
0.5565
Epoch 57: val_loss did not improve from 0.97569
1559/1559 [==============================] - 79s 51ms/step - loss: 1.0437 -
accuracy: 0.5565 - val_loss: 1.0409 - val_accuracy: 0.5575
Epoch 58/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0409 - accuracy:
0.5585
Epoch 58: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.0409 -
accuracy: 0.5585 - val_loss: 1.0353 - val_accuracy: 0.5575
Epoch 59/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0377 - accuracy:
0.5600
Epoch 59: val_loss did not improve from 0.97569
1559/1559 [==============================] - 80s 51ms/step - loss: 1.0377 -
accuracy: 0.5600 - val_loss: 1.0418 - val_accuracy: 0.5600
Epoch 60/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0361 - accuracy:
0.5610
Epoch 60: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.0362 -
accuracy: 0.5610 - val_loss: 1.0431 - val_accuracy: 0.5555
Epoch 61/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0336 - accuracy:
0.5624
Epoch 61: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.0336 -
accuracy: 0.5624 - val_loss: 1.0341 - val_accuracy: 0.5598
```

```
Epoch 62/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0313 - accuracy:
0.5634
Epoch 62: val_loss did not improve from 0.97569
1559/1559 [==============================] - 79s 51ms/step - loss: 1.0313 -
accuracy: 0.5635 - val_loss: 1.0289 - val_accuracy: 0.5608
Epoch 63/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0298 - accuracy:
0.5640
Epoch 63: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.0298 -
accuracy: 0.5640 - val_loss: 1.0375 - val_accuracy: 0.5597
Epoch 64/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0278 - accuracy:
0.5650
Epoch 64: val_loss did not improve from 0.97569
1559/1559 [==============================] - 79s 50ms/step - loss: 1.0278 -
accuracy: 0.5650 - val_loss: 1.0230 - val_accuracy: 0.5640
Epoch 65/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0265 - accuracy:
0.5654
Epoch 65: val_loss did not improve from 0.97569
1559/1559 [==============================] - 79s 51ms/step - loss: 1.0265 -
accuracy: 0.5654 - val_loss: 1.0358 - val_accuracy: 0.5616
Epoch 66/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0253 - accuracy:
0.5666
Epoch 66: val_loss did not improve from 0.97569
1559/1559 [==============================] - 79s 51ms/step - loss: 1.0252 -
accuracy: 0.5666 - val_loss: 1.0381 - val_accuracy: 0.5572
Epoch 67/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0232 - accuracy:
0.5681
Epoch 67: val_loss did not improve from 0.97569
1559/1559 [==============================] - 78s 50ms/step - loss: 1.0232 -
accuracy: 0.5681 - val_loss: 1.0224 - val_accuracy: 0.5657
Epoch 68/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0212 - accuracy:
0.5680
Epoch 68: val_loss did not improve from 0.97569
1559/1559 [==============================] - 79s 51ms/step - loss: 1.0212 -
accuracy: 0.5680 - val_loss: 1.0273 - val_accuracy: 0.5659
Epoch 69/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0202 - accuracy:
0.5693
Epoch 69: val_loss did not improve from 0.97569
1559/1559 [==============================] - 79s 51ms/step - loss: 1.0202 -
accuracy: 0.5693 - val_loss: 1.0314 - val_accuracy: 0.5644
```

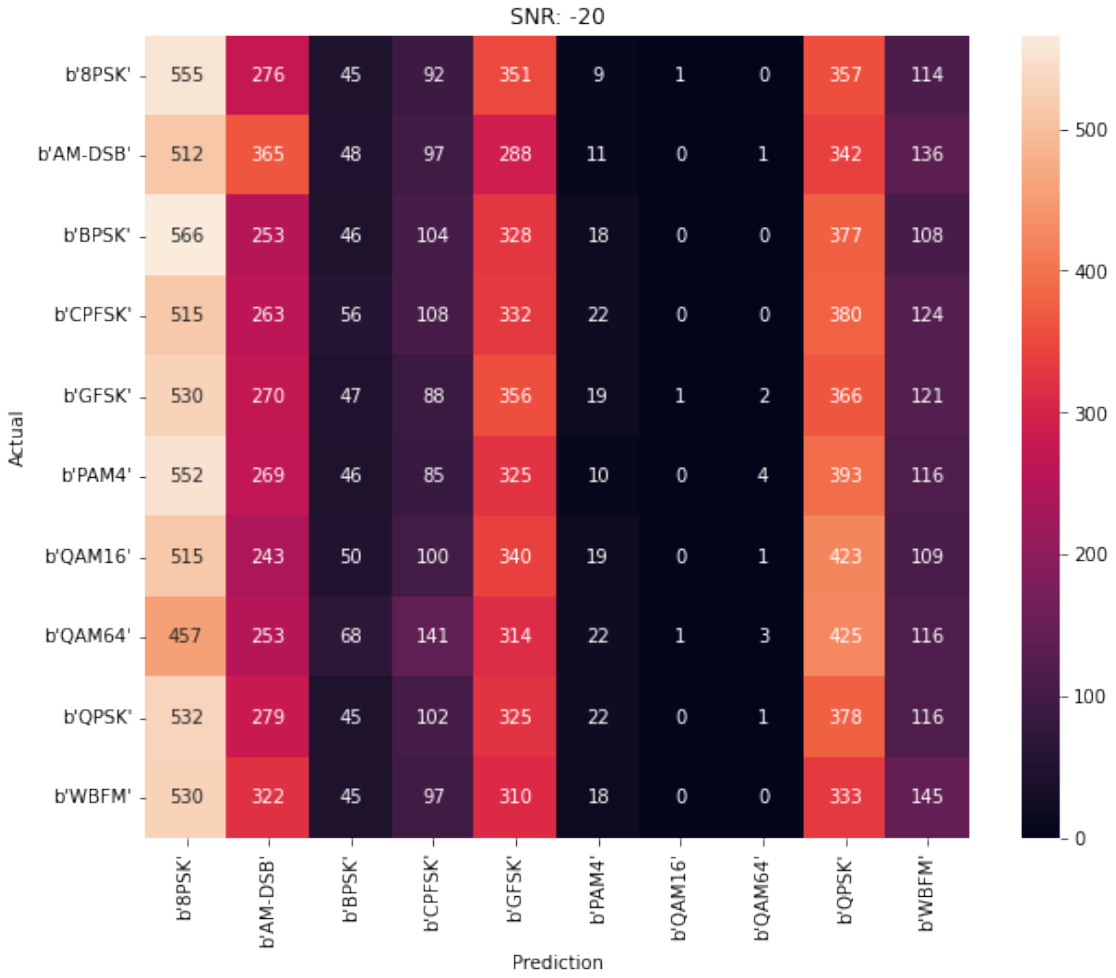
```
Epoch 70/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0180 - accuracy:
0.5698
Epoch 70: val_loss did not improve from 0.97569
1559/1559 [==============================] - 77s 50ms/step - loss: 1.0180 -
accuracy: 0.5698 - val_loss: 1.0235 - val_accuracy: 0.5650
Epoch 71/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0170 - accuracy:
0.5708
Epoch 71: val_loss did not improve from 0.97569
1559/1559 [==============================] - 79s 51ms/step - loss: 1.0169 -
accuracy: 0.5708 - val_loss: 1.0239 - val_accuracy: 0.5654
Epoch 72/200
1558/1559 [============================>.] - ETA: 0s - loss: 1.0152 - accuracy:
0.5709
Epoch 72: val_loss did not improve from 0.97569
1559/1559 [==============================] - 79s 51ms/step - loss: 1.0152 -
accuracy: 0.5709 - val_loss: 1.0349 - val_accuracy: 0.5586
```

[ ]:
```
plot_model_history(history, 'CNN Model')
model_scoring(cnn_lstm_model_2, X_test, testing_pair_labels)
```

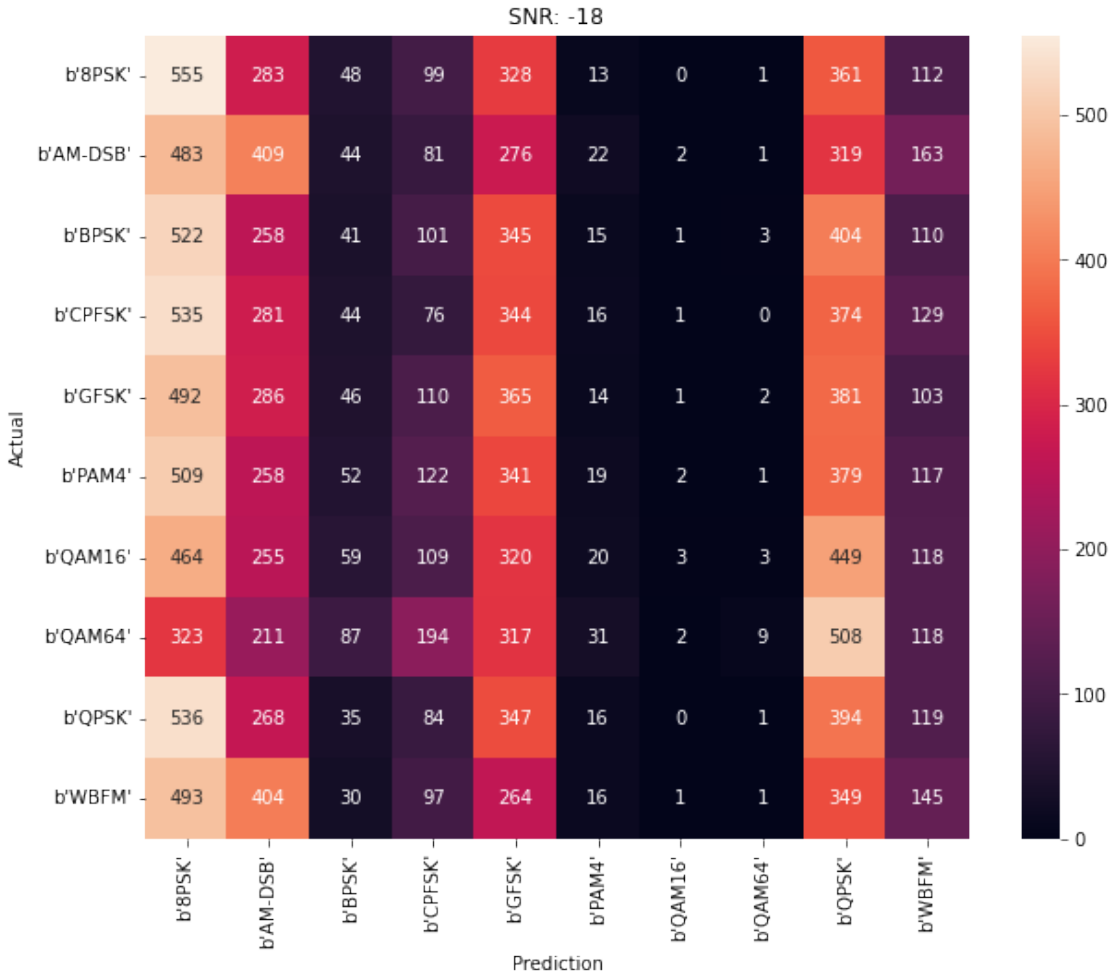



```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
```

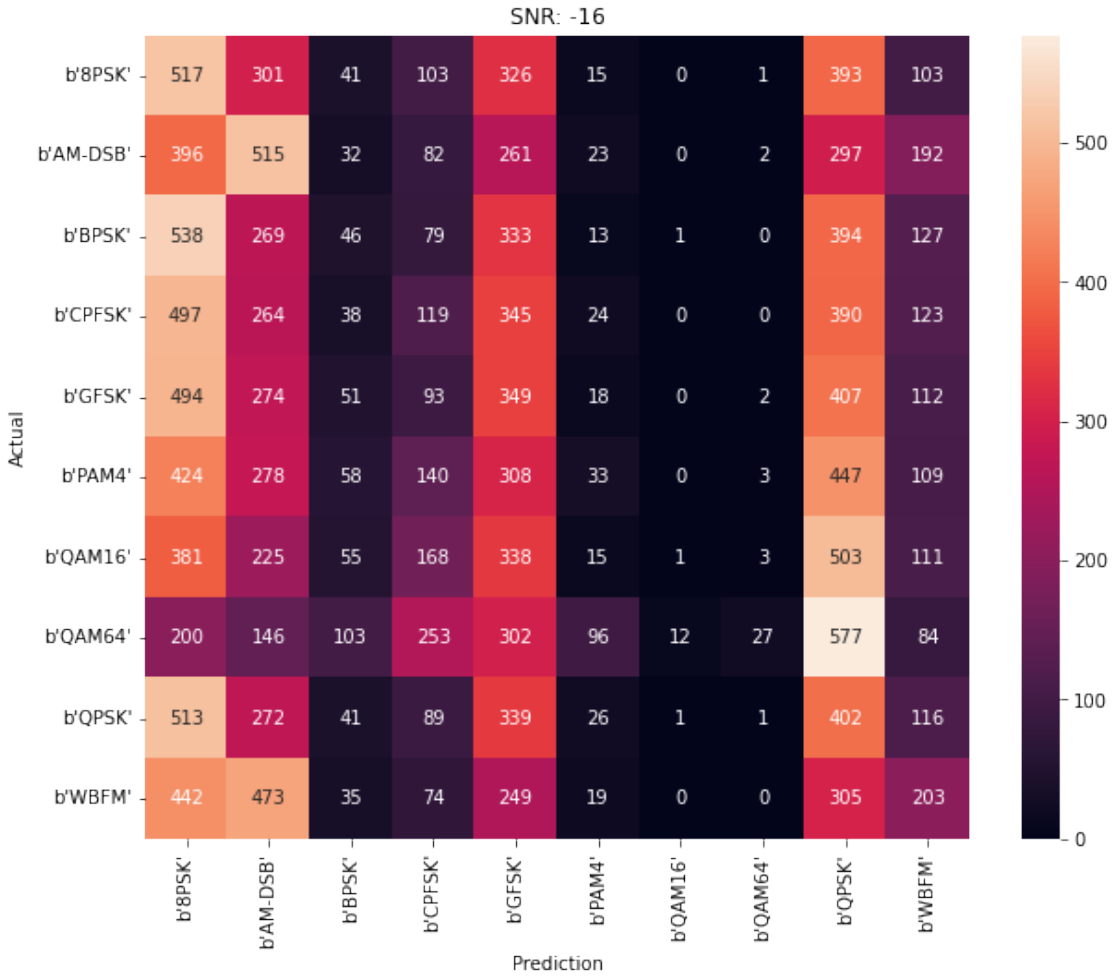
```
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = -20 is 0.10922222222222222%
```



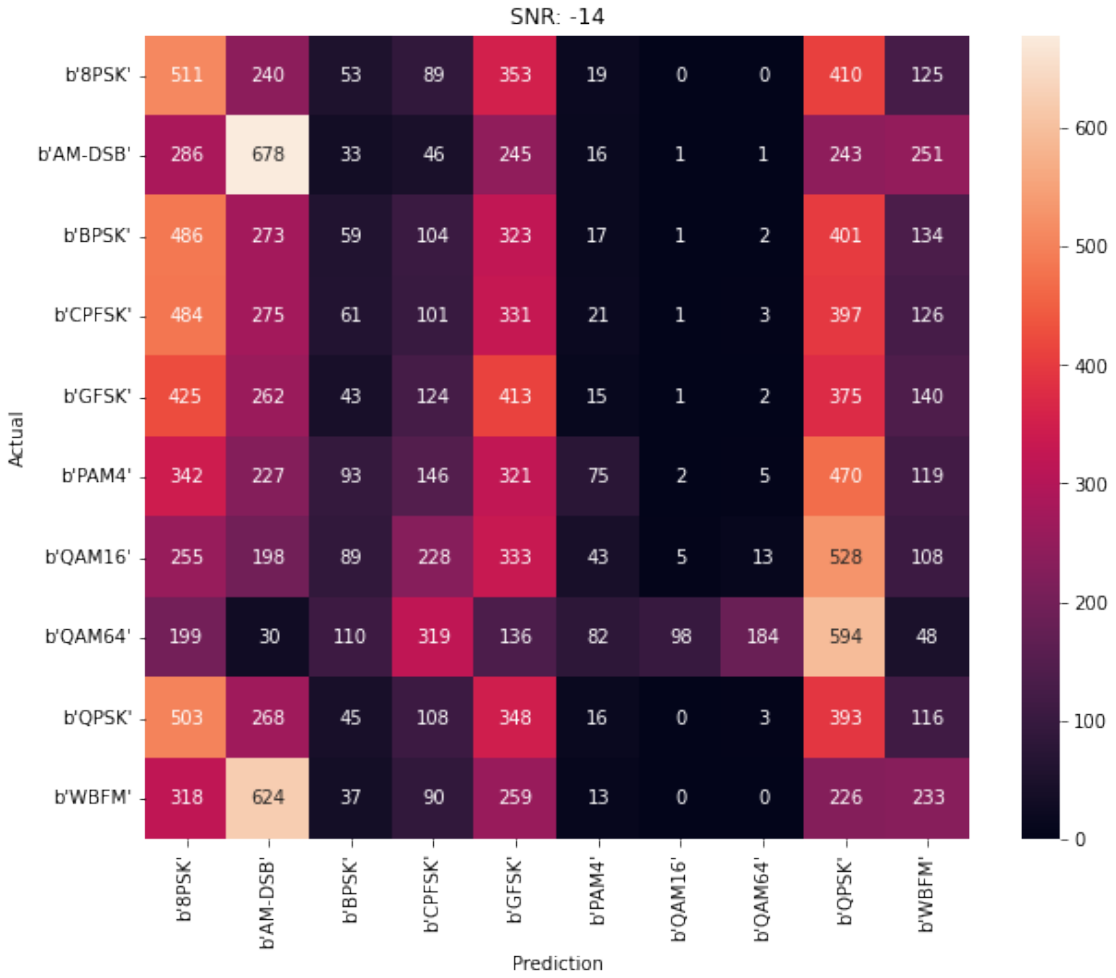

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = -18 is 0.112%
```



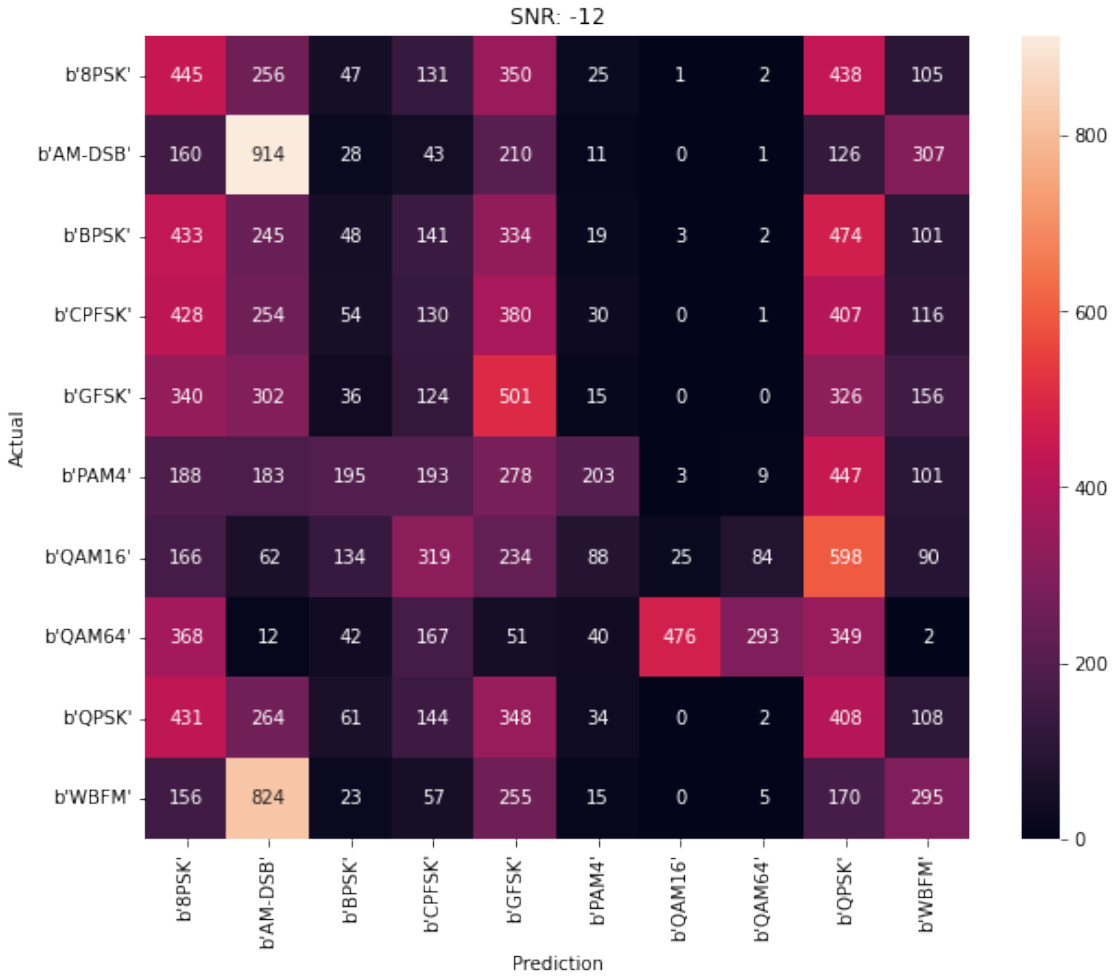

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = -16 is 0.1228888888888889%
```



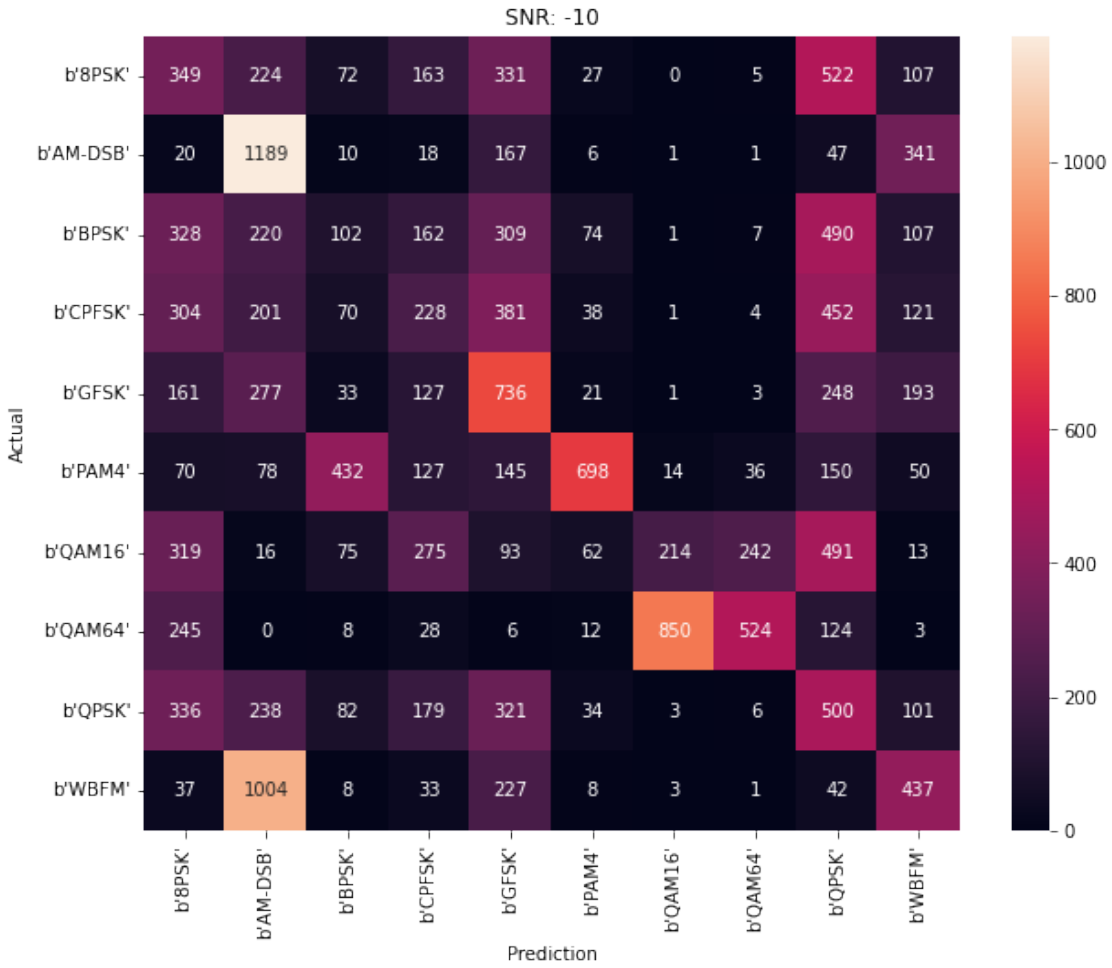

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = -14 is 0.14733333333333334%
```



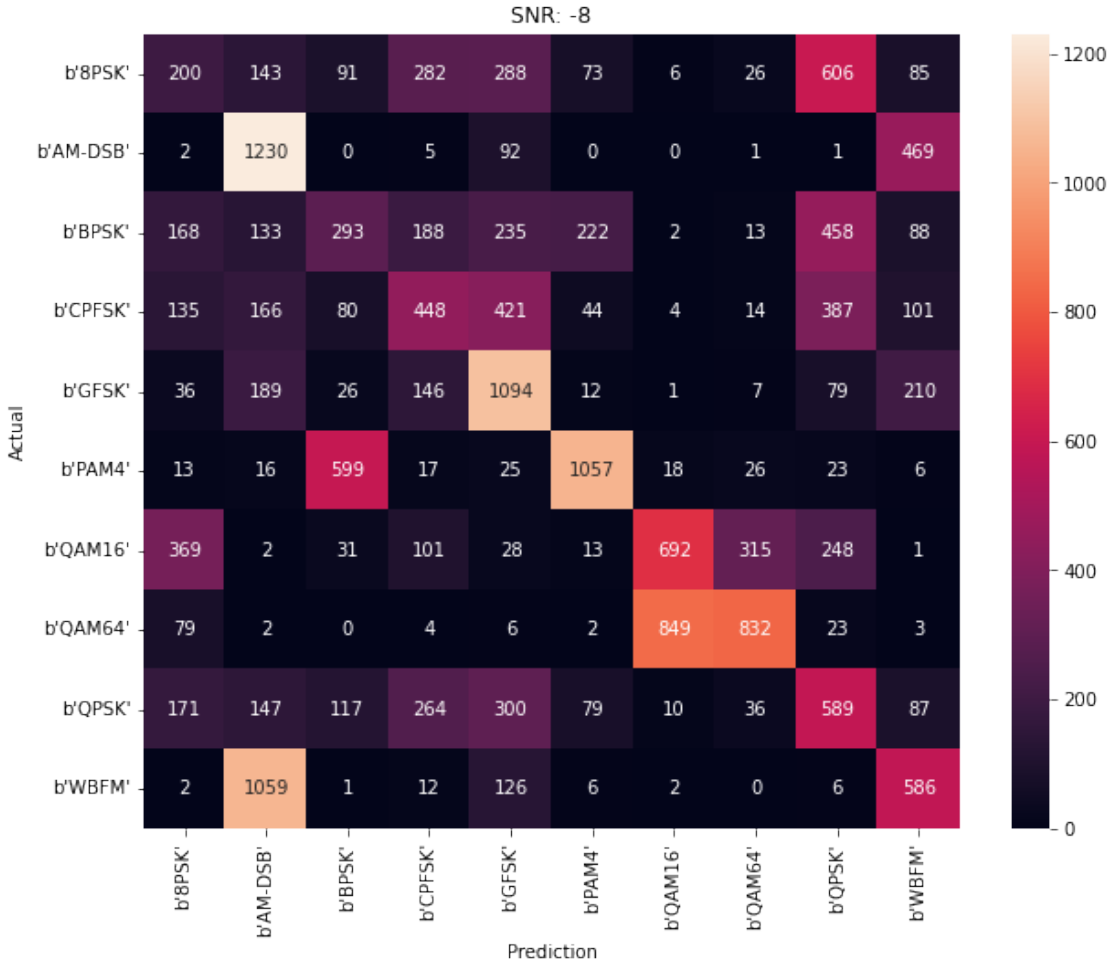

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = -12 is 0.18122222222222223%
```



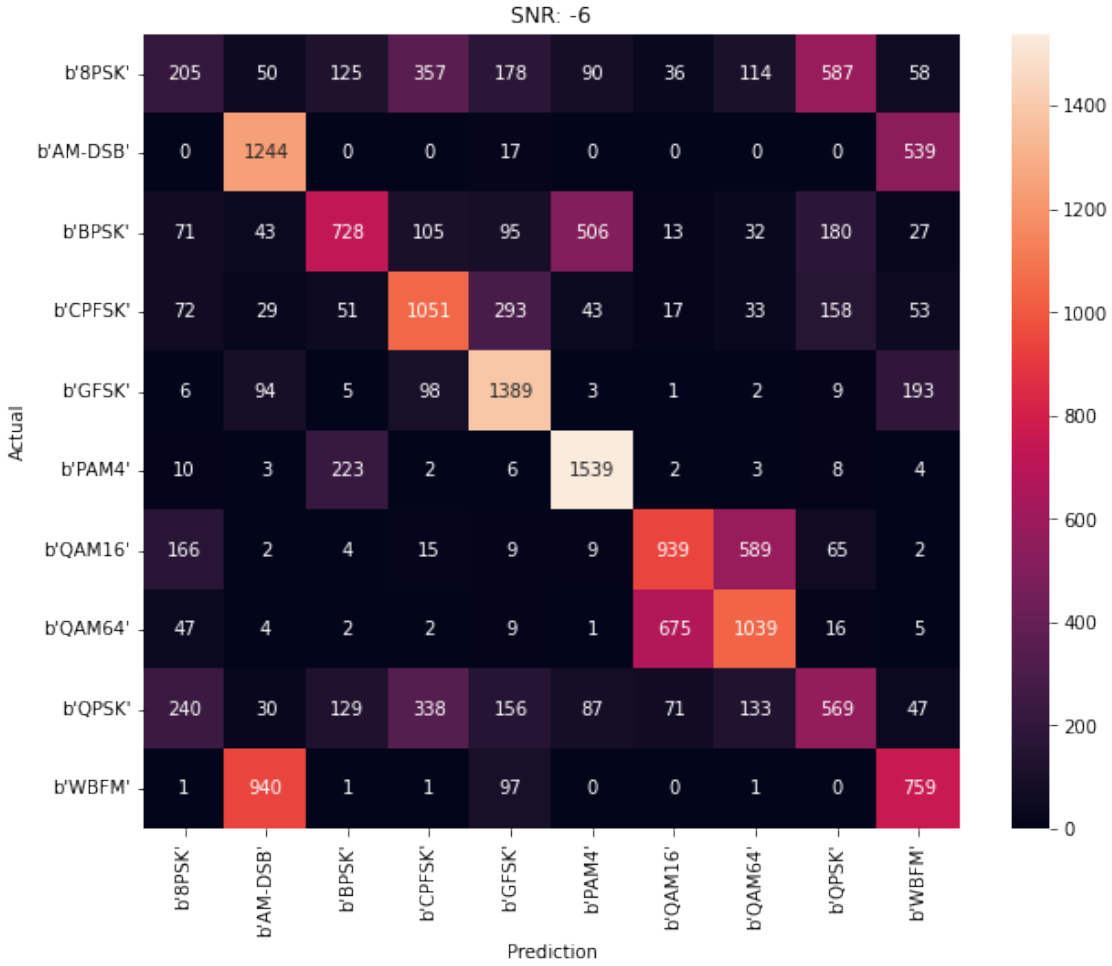

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = -10 is 0.2765%
```



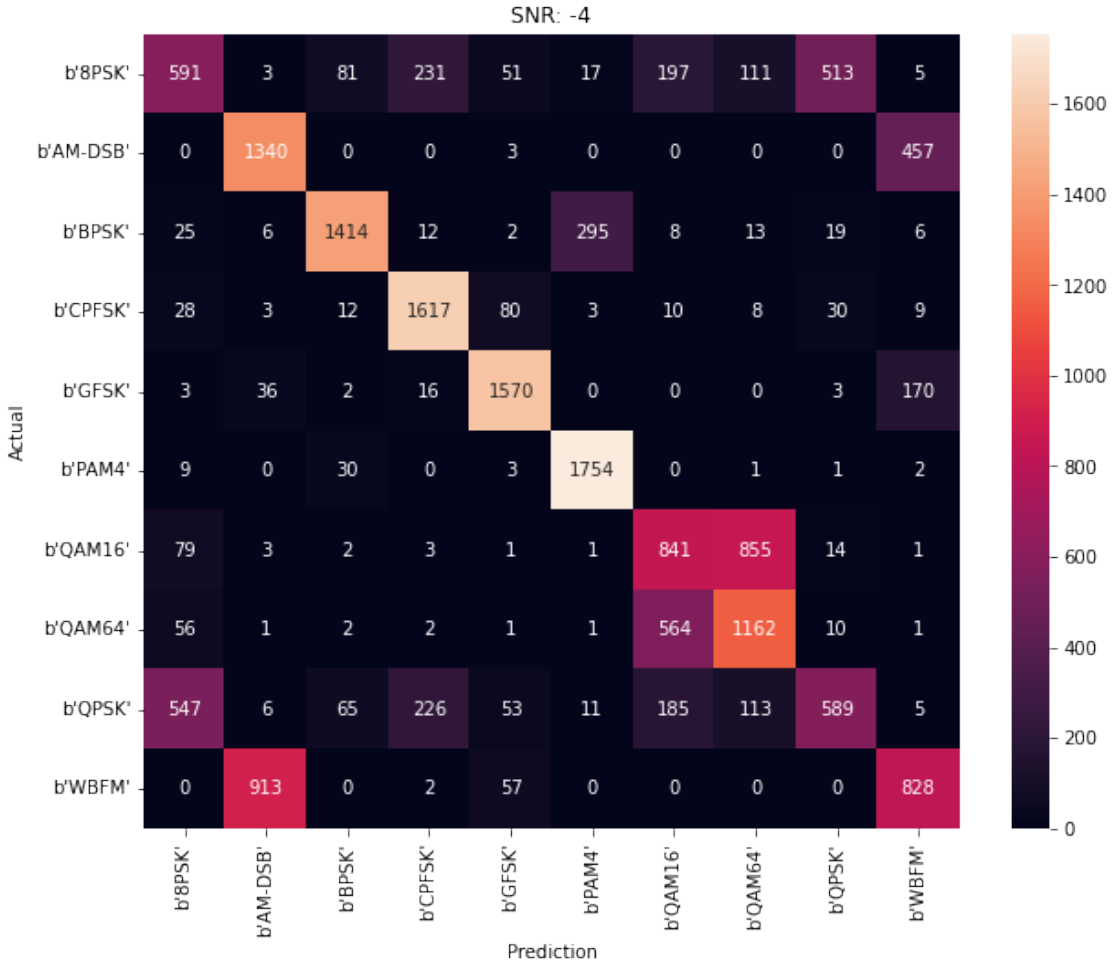

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = -8 is 0.39005555555555554%
```



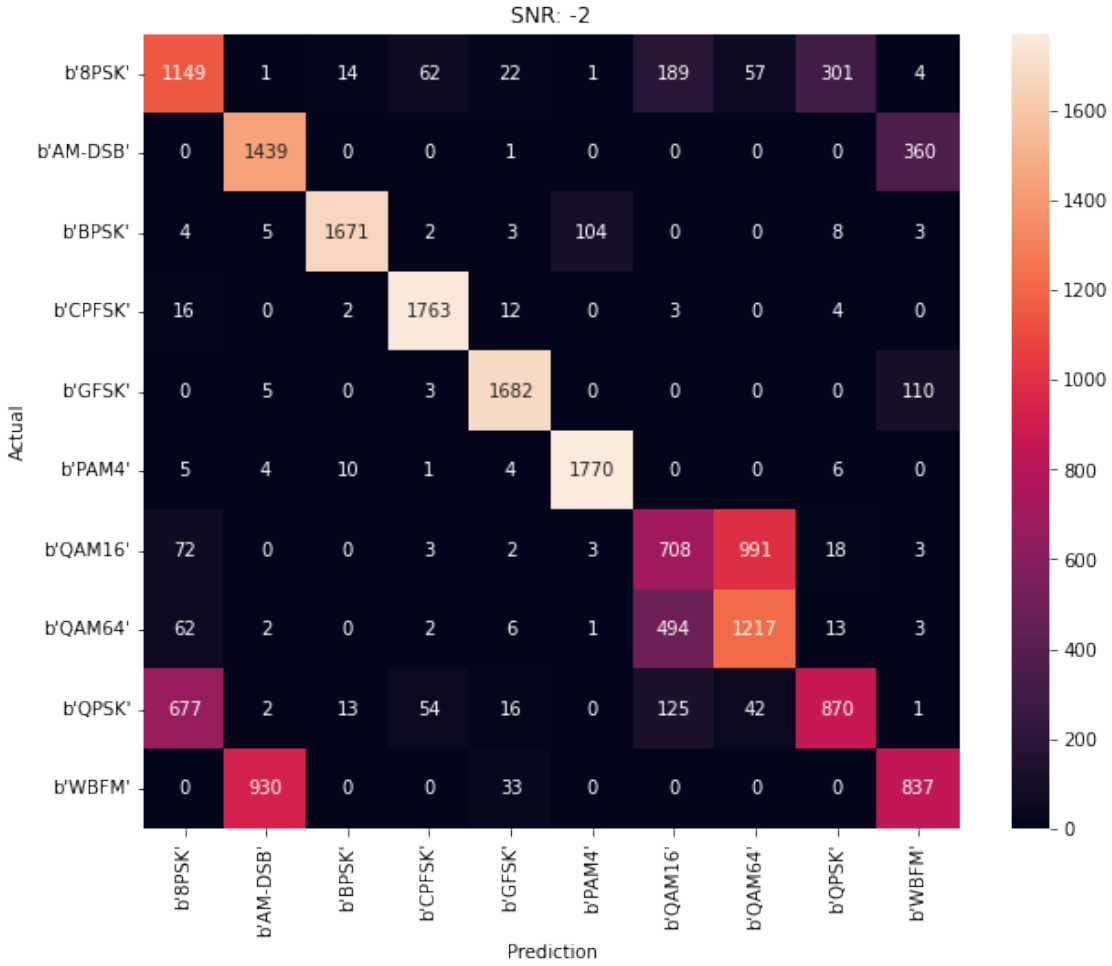

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = -6 is 0.5256666666666666%
```



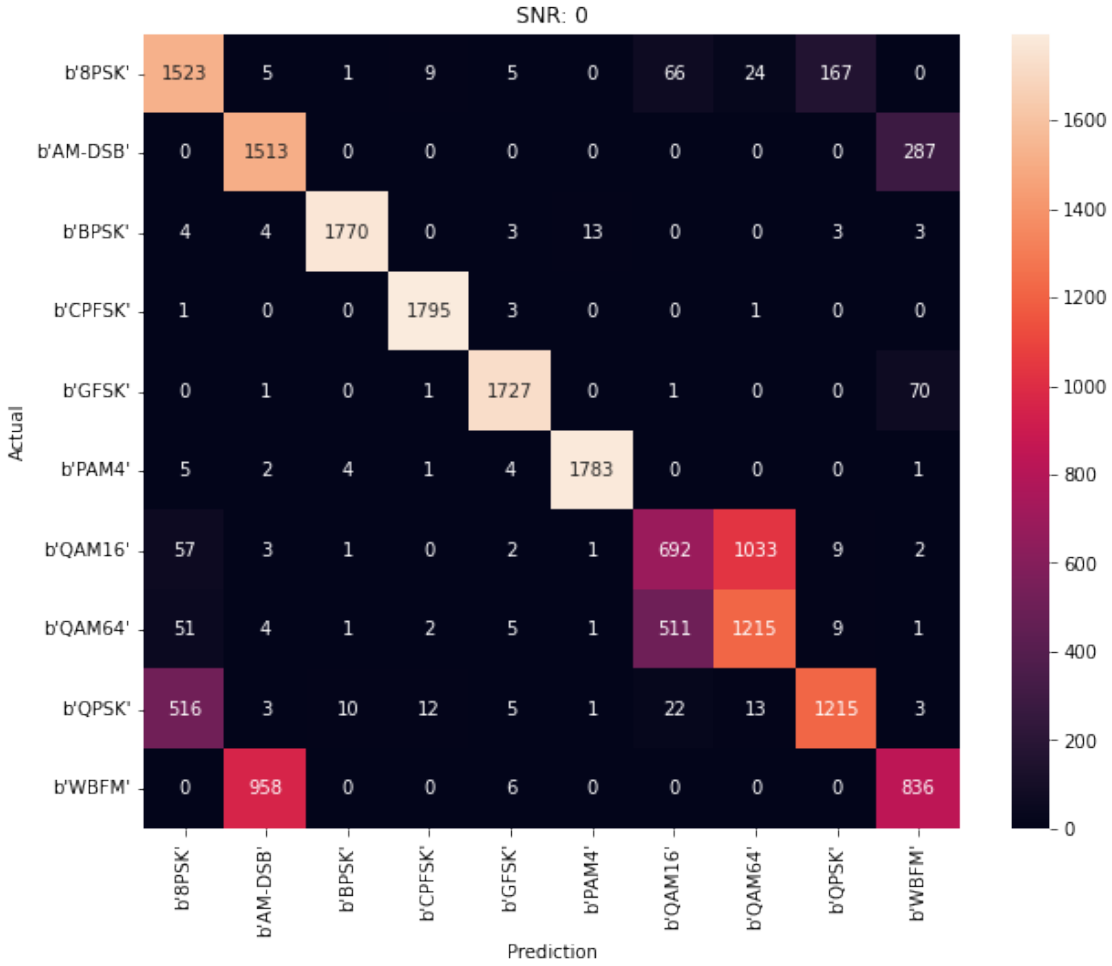

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = -4 is 0.6503333333333333%
```



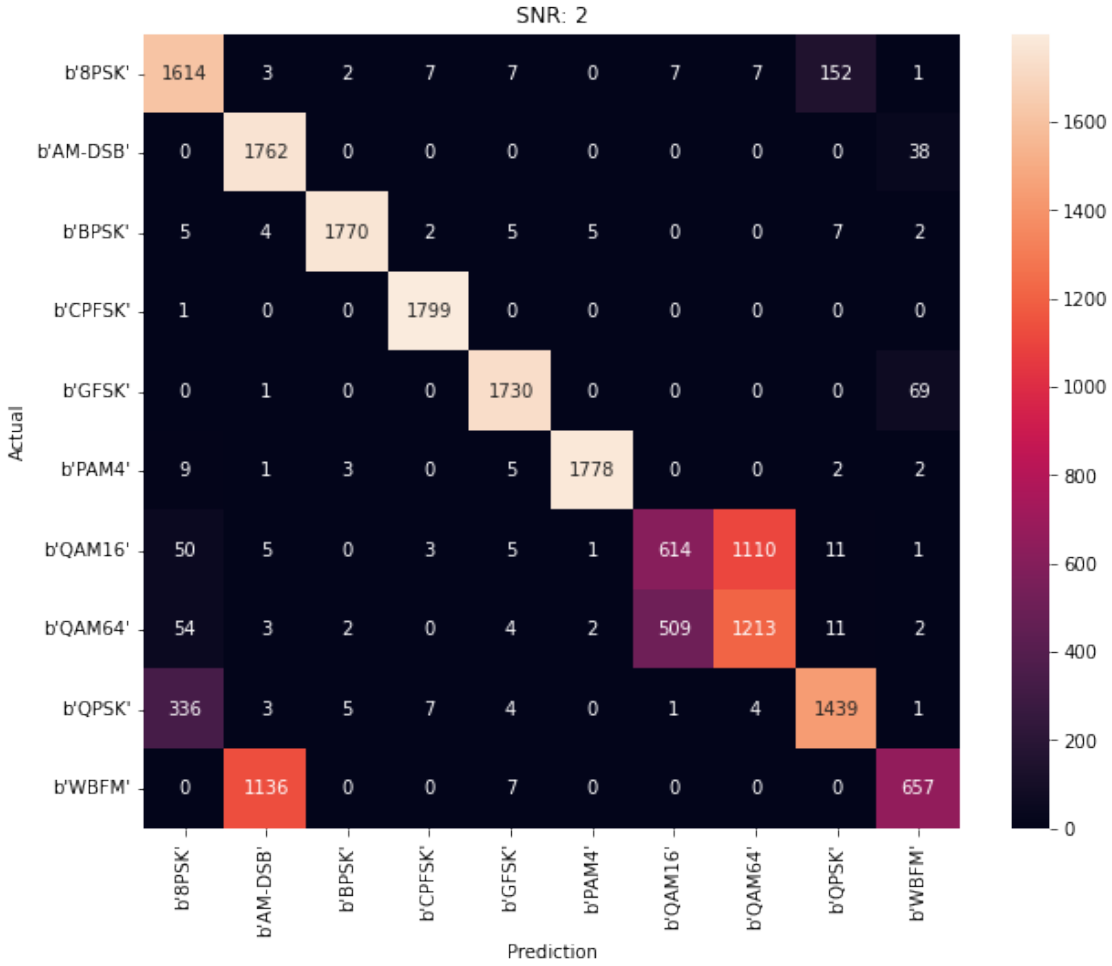

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = -2 is 0.7281111111111112%
```



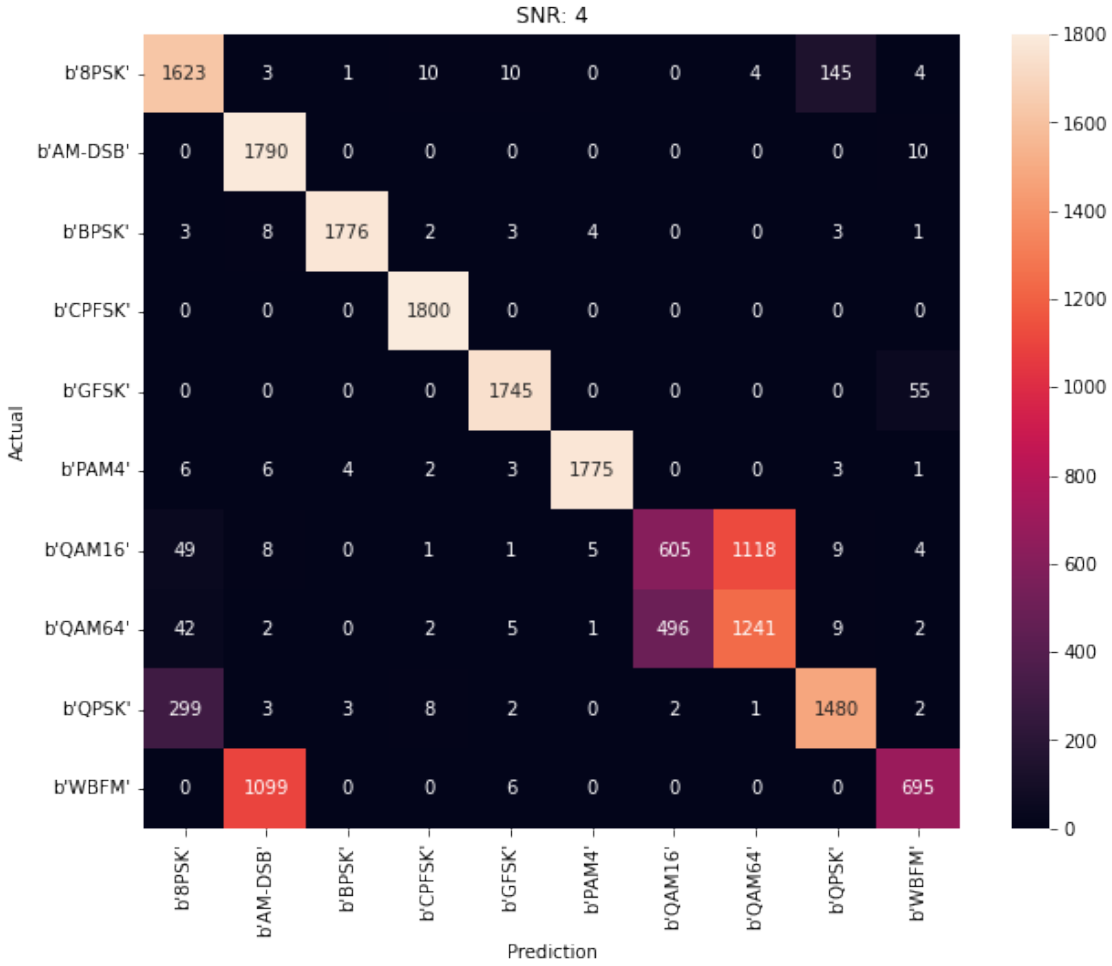

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = 0 is 0.7816111111111111%
```



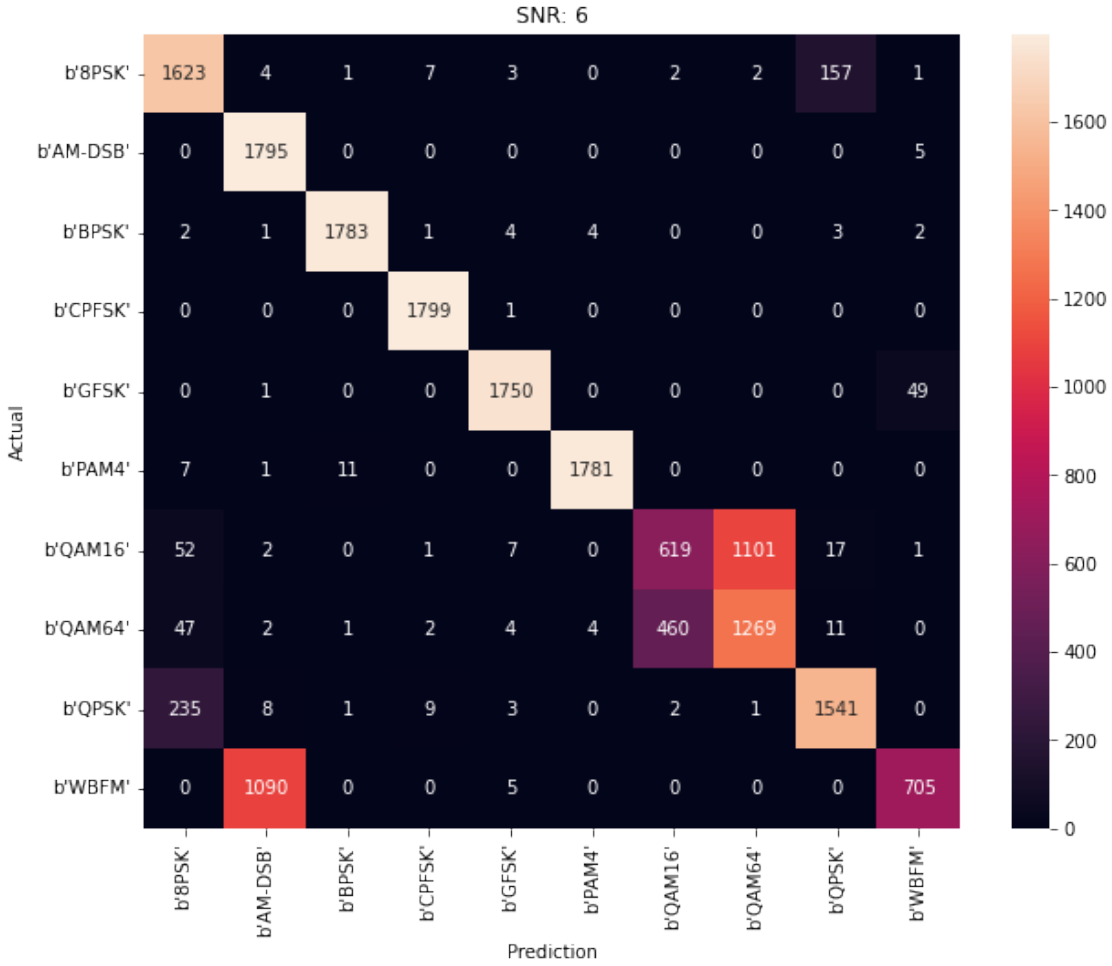

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = 2 is 0.7986666666666666%
```



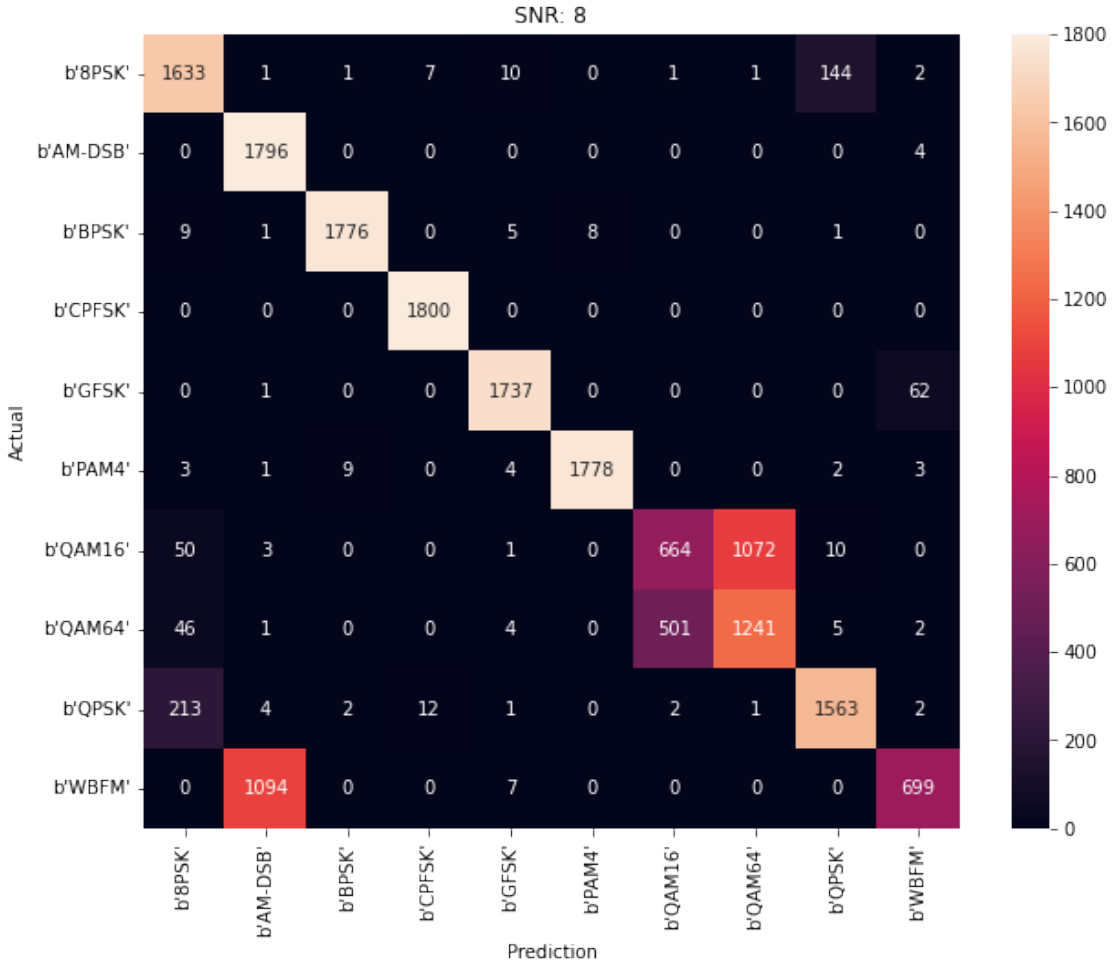

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = 4 is 0.8072222222222222%
```



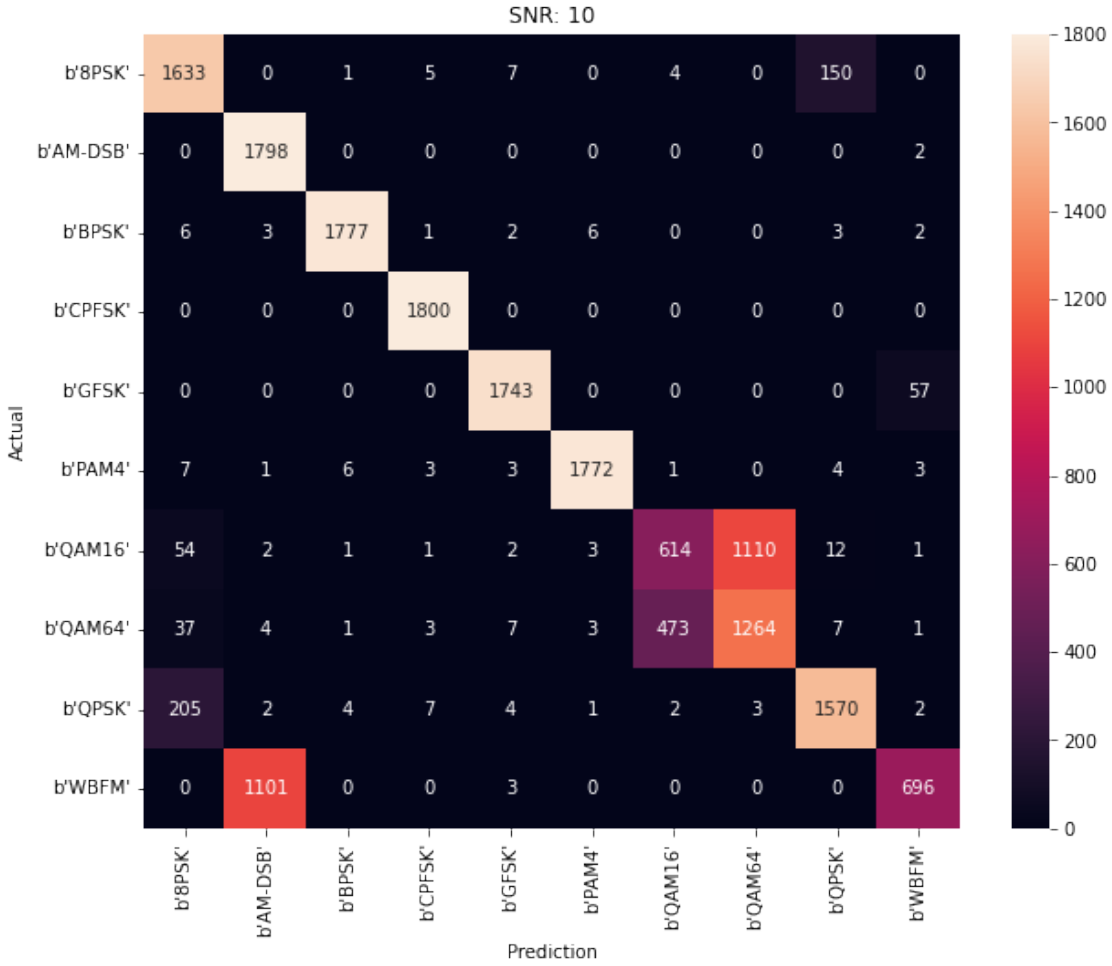

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = 6 is 0.8147222222222222%
```



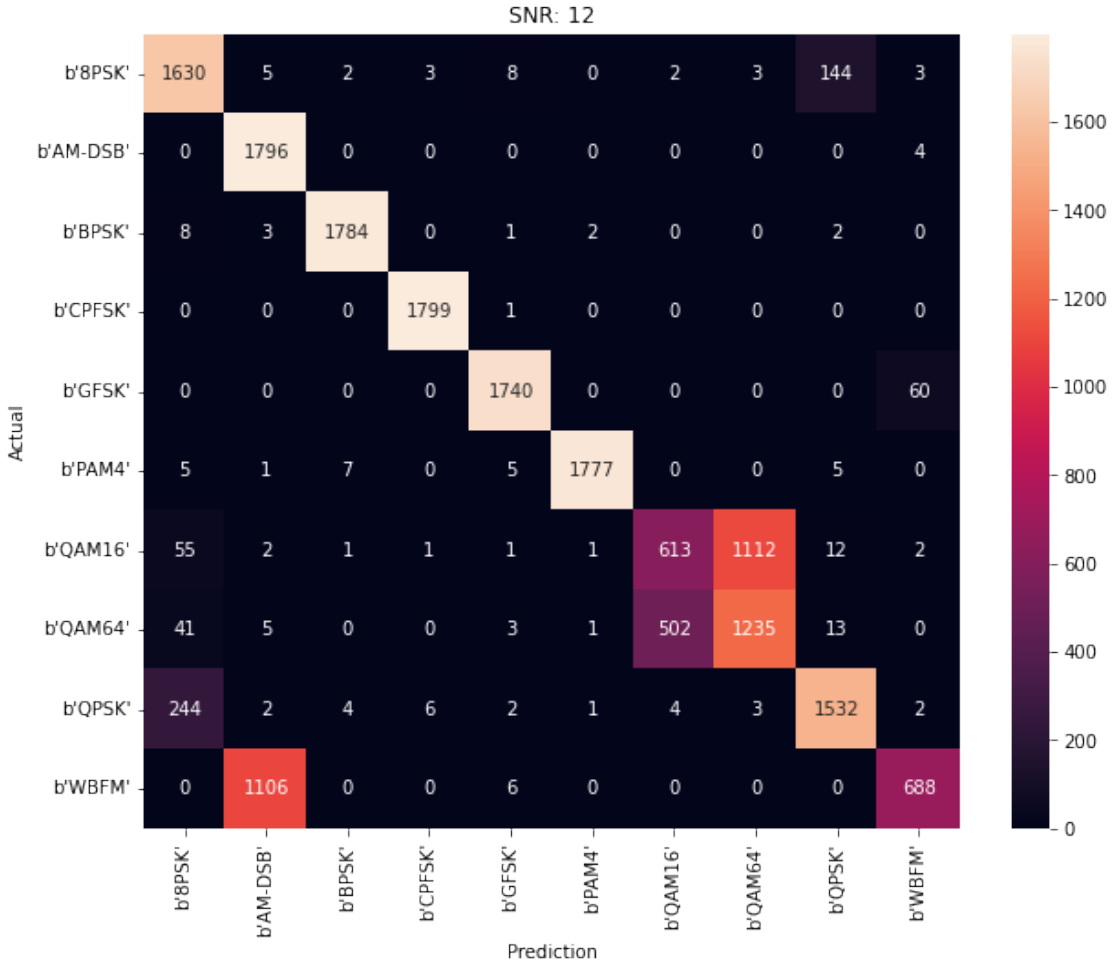

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = 8 is 0.8159444444444445%
```



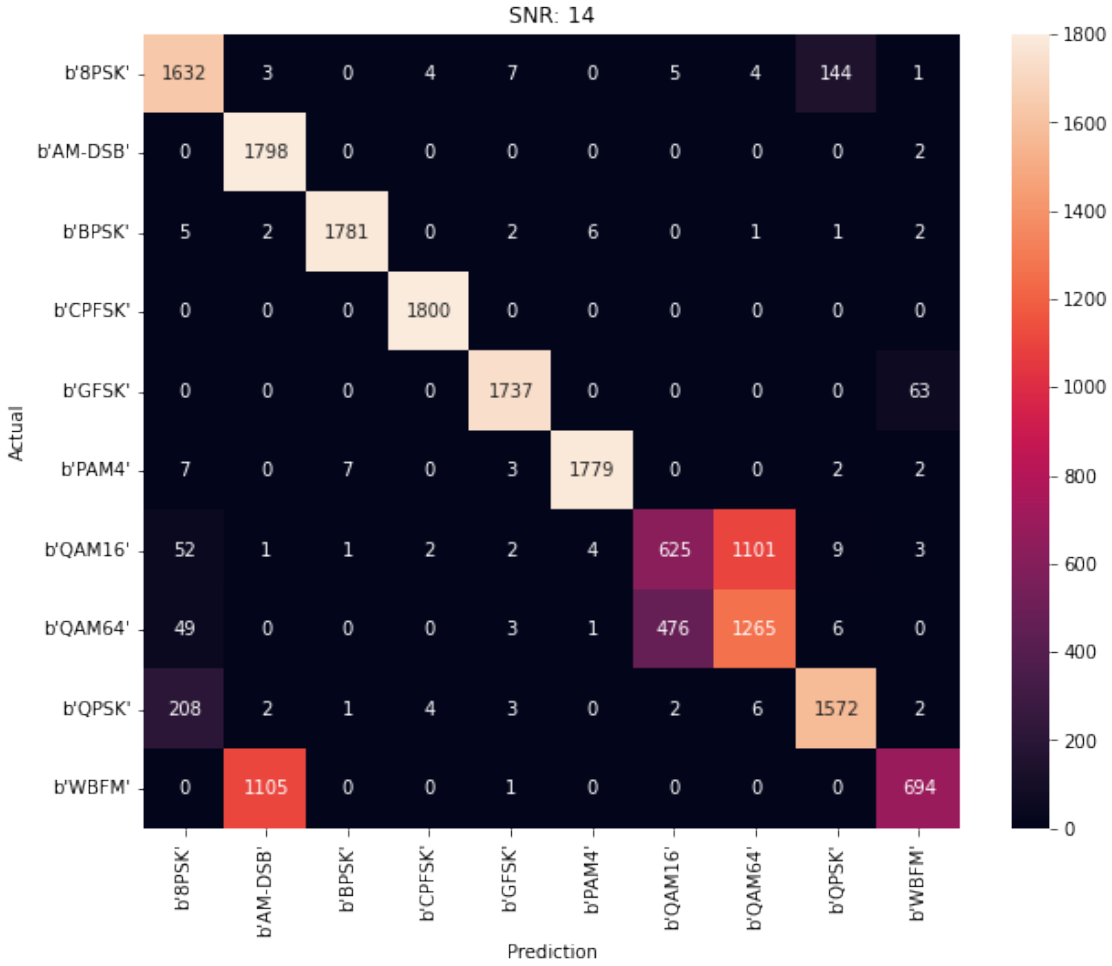

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = 10 is 0.8148333333333333%
```



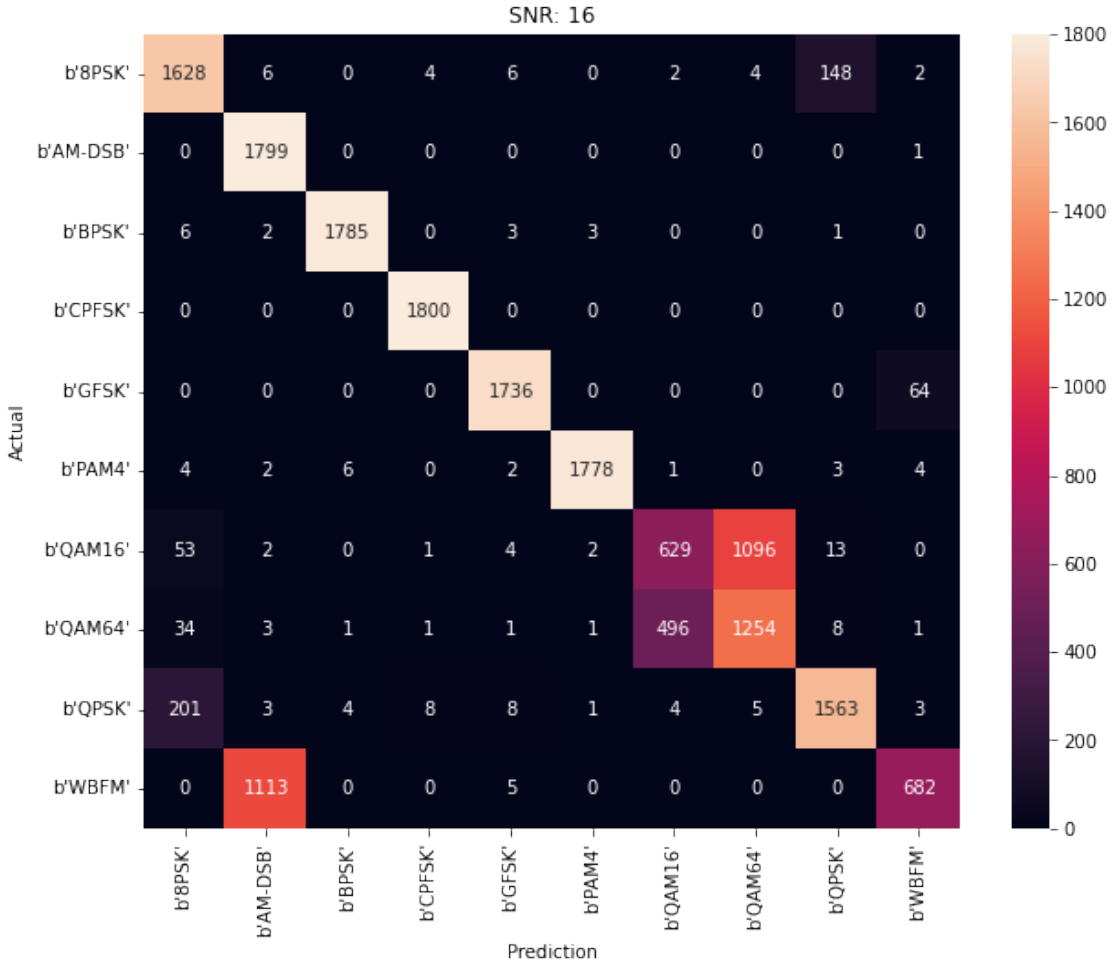

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = 12 is 0.8107777777777778%
```



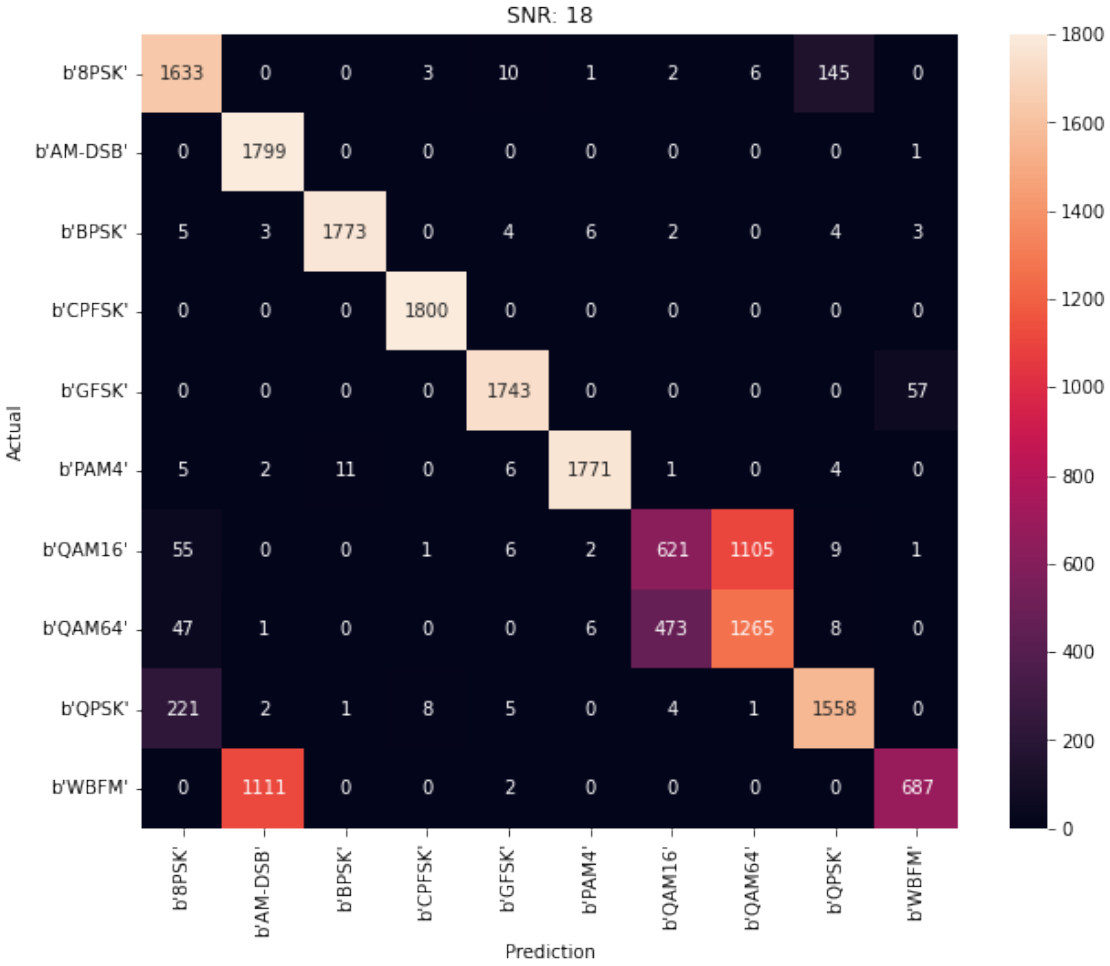

```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = 14 is 0.8157222222222222%
```

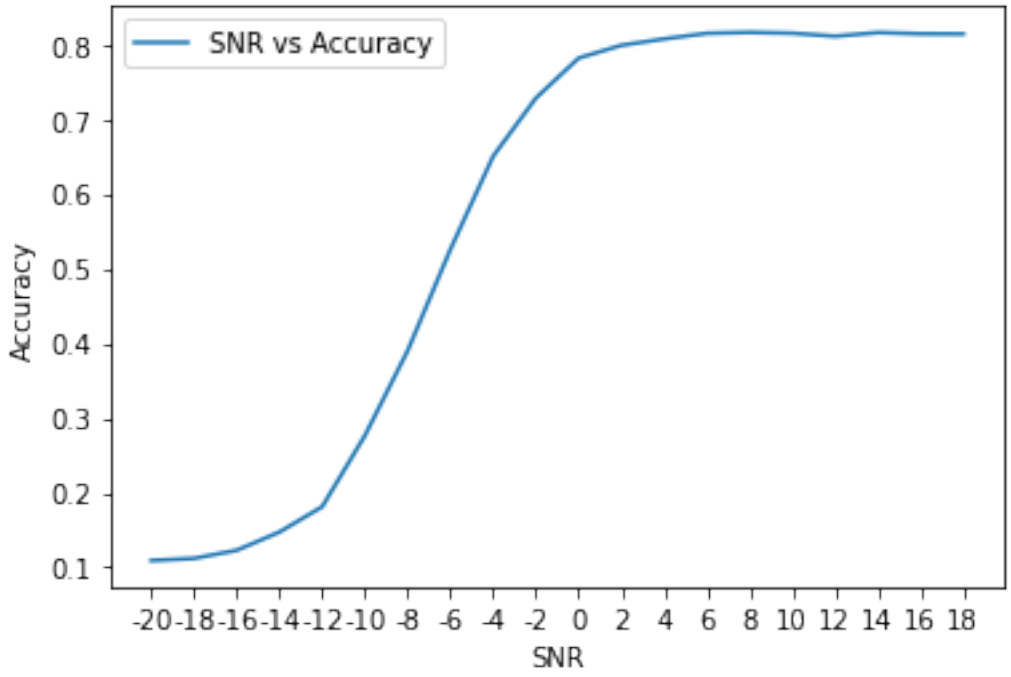



```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = 16 is 0.8141111111111111%
```




```
/usr/local/lib/python3.7/dist-
packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning:
Even though the `tf.config.experimental_run_functions_eagerly` option is set,
this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  "Even though the `tf.config.experimental_run_functions_eagerly` "

Accuracy at SNR = 18 is 0.8138888888888889%
```


SNR: 18
b'8PSK' 1633 0 0 3 10 1 2 6 145 0
b'AM-DSB' 0 1799 0 0 0 0 0 0 0 1
b'BPSK' 5 3 1773 0 4 6 2 0 4 3
b'CPFSK' 0 0 0 1800 0 0 0 0 0 0
b'GFSK' 0 0 0 0 1743 0 0 0 0 57
b'PAM4' 5 2 11 0 6 1771 1 0 4 0
b'QAM16' 55 0 0 1 6 2 621 1105 9 1
b'QAM64' 47 1 0 0 0 6 473 1265 8 0
b'QPSK' 221 2 1 8 5 0 4 1 1558 0
b'WBFM' 0 1111 0 0 2 0 0 0 0 687
Actual
b'8PSK' b'AM-DSB' b'BPSK' b'CPFSK' b'GFSK' b'PAM4' b'QAM16' b'QAM64' b'QPSK' b'WBFM'
Prediction
1800 1600 1400 1200 1000 800 600 400 200 0



[ ]:

[ ]:

[ ]:

7.1 CNN Model

8 Combined Feature Space

[19]:
```
Combined_training_data = np.concatenate((training_data ,fdit_training_data,␣
 ↪fiit_training_data), axis=2)
Combined_validation_data = np.concatenate((validation_data␣
 ↪,fdit_validation_data, fiit_validation_data), axis=2)
Combined_testing_data = np.concatenate((testing_data ,fdit_testing_data,␣
 ↪fiit_testing_data), axis=2)
```

[20]:
```
print('combined training data shape:', Combined_training_data.shape)
print('combined validation data shape:',Combined_validation_data.shape)
print('combined testing data shape:', Combined_testing_data.shape)
```

```
combined training data shape: (798000, 2, 384)
combined validation data shape: (42000, 2, 384)
combined testing data shape: (360000, 2, 384)
```

[21]:
```
del training_data
del fdit_training_data
del fiit_training_data

del validation_data
del fdit_validation_data
del fiit_validation_data

del testing_data
del fdit_testing_data
del fiit_testing_data
```

[22]:
```
X_trainp = np.asarray(np.transpose(Combined_training_data, axes=(0,2,1)))
X_valp   = np.asarray(np.transpose(Combined_validation_data  , axes=(0,2,1)))
n_timesteps, n_features, n_outputs = X_trainp.shape[1], X_trainp.shape[2],␣
 ↪validation_onehot.shape[1]
n_steps, n_length = 4, 32
```

[44]:
```
import gc, torch
for _ in range(0,300):
  gc.collect()
  torch.cuda.empty_cache()
```

[ ]:
```
print(X_trainp.shape)
```

```
(798000, 384, 2)
```

[23]:
```
X_trainp = X_trainp.reshape((X_trainp.shape[0], n_steps, n_length, n_features*3))
print(X_trainp.shape)
```

```
(798000, 4, 32, 6)
```

[25]:
```
del Combined_training_data
```

[27]:
```
X_valp = X_valp.reshape((X_valp.shape[0], n_steps, n_length, n_features*3))
```

[28]:
```
del Combined_validation_data
```

[33]:
```
X_test = np.asarray(np.transpose(Combined_testing_data, axes=(0,2,1)))
n_timesteps, n_features, n_outputs = X_test.shape[1], X_test.shape[2],␣
 ↪validation_onehot.shape[1]
n_steps, n_length = 4, 32
X_test = X_test.reshape((X_test.shape[0], n_steps, n_length, n_features*3))
```

[34]:
```
del Combined_testing_data
```

[36]:
```
es = tf.keras.callbacks.EarlyStopping(monitor="val_loss", patience=5,␣
↪restore_best_weights=True,)
checkpointer = ModelCheckpoint(filepath='saved_models/cnn_lstm_classification.
↪hdf5', verbose=1, save_best_only=True)
```

[37]:
```
cnn_lstm_model_2 = Sequential()
cnn_lstm_model_2.add(TimeDistributed(Conv1D(filters=256,  padding =␣
↪'same',kernel_size=3, activation='relu'),␣
↪input_shape=(None,n_length,n_features*3)))
cnn_lstm_model_2.add(TimeDistributed(Dropout(0.5)))
cnn_lstm_model_2.add(TimeDistributed(Conv1D(filters=64,  padding = 'same'␣
↪,kernel_size=3, activation='relu')))
cnn_lstm_model_2.add(TimeDistributed(Dropout(0.5)))
cnn_lstm_model_2.add(TimeDistributed(Flatten()))
cnn_lstm_model_2.add(LSTM(100))
cnn_lstm_model_2.add(Dense(128, activation='relu'))
cnn_lstm_model_2.add(Dense(n_outputs, activation='softmax'))
cnn_lstm_model_2.compile(loss=tf.keras.losses.CategoricalCrossentropy(),␣
↪metrics=['accuracy'], optimizer=tf.keras.optimizers.
↪Adam(learning_rate=learning_rate))
```

[38]:
```
with tf.device('/device:GPU:0'):
  history = cnn_lstm_model_2.fit(X_trainp, training_onehot, batch_size=512,␣
↪epochs=epochs, validation_data=(X_valp, validation_onehot), callbacks=[es,␣
↪checkpointer], verbose=1)
```

```
Epoch 1/200
1559/1559 [==============================] - ETA: 0s - loss: 1.4985 - accuracy:
0.3643
Epoch 1: val_loss improved from inf to 1.35977, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 66s 33ms/step - loss: 1.4985 -
accuracy: 0.3643 - val_loss: 1.3598 - val_accuracy: 0.4056
Epoch 2/200
1559/1559 [==============================] - ETA: 0s - loss: 1.2907 - accuracy:
0.4437
Epoch 2: val_loss improved from 1.35977 to 1.15379, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 49s 31ms/step - loss: 1.2907 -
accuracy: 0.4437 - val_loss: 1.1538 - val_accuracy: 0.5043
Epoch 3/200
1559/1559 [==============================] - ETA: 0s - loss: 1.1613 - accuracy:
0.4983
Epoch 3: val_loss improved from 1.15379 to 1.12273, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 49s 32ms/step - loss: 1.1613 -
accuracy: 0.4983 - val_loss: 1.1227 - val_accuracy: 0.5161
```

```
Epoch 4/200
1559/1559 [==============================] - ETA: 0s - loss: 1.1364 - accuracy:
0.5093
Epoch 4: val_loss improved from 1.12273 to 1.10478, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 49s 32ms/step - loss: 1.1364 -
accuracy: 0.5093 - val_loss: 1.1048 - val_accuracy: 0.5196
Epoch 5/200
1559/1559 [==============================] - ETA: 0s - loss: 1.1190 - accuracy:
0.5184
Epoch 5: val_loss improved from 1.10478 to 1.09756, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 49s 32ms/step - loss: 1.1190 -
accuracy: 0.5184 - val_loss: 1.0976 - val_accuracy: 0.5325
Epoch 6/200
1559/1559 [==============================] - ETA: 0s - loss: 1.1059 - accuracy:
0.5253
Epoch 6: val_loss improved from 1.09756 to 1.09750, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 49s 31ms/step - loss: 1.1059 -
accuracy: 0.5253 - val_loss: 1.0975 - val_accuracy: 0.5305
Epoch 7/200
1559/1559 [==============================] - ETA: 0s - loss: 1.0963 - accuracy:
0.5306
Epoch 7: val_loss improved from 1.09750 to 1.08127, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 49s 31ms/step - loss: 1.0963 -
accuracy: 0.5306 - val_loss: 1.0813 - val_accuracy: 0.5369
Epoch 8/200
1559/1559 [==============================] - ETA: 0s - loss: 1.0879 - accuracy:
0.5351
Epoch 8: val_loss improved from 1.08127 to 1.07969, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 49s 32ms/step - loss: 1.0879 -
accuracy: 0.5351 - val_loss: 1.0797 - val_accuracy: 0.5363
Epoch 9/200
1559/1559 [==============================] - ETA: 0s - loss: 1.0791 - accuracy:
0.5398
Epoch 9: val_loss improved from 1.07969 to 1.04920, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 50s 32ms/step - loss: 1.0791 -
accuracy: 0.5398 - val_loss: 1.0492 - val_accuracy: 0.5523
Epoch 10/200
1559/1559 [==============================] - ETA: 0s - loss: 1.0701 - accuracy:
0.5454
Epoch 10: val_loss did not improve from 1.04920
1559/1559 [==============================] - 49s 31ms/step - loss: 1.0701 -
accuracy: 0.5454 - val_loss: 1.0548 - val_accuracy: 0.5543
```

```
Epoch 11/200
1559/1559 [==============================] - ETA: 0s - loss: 1.0626 - accuracy:
0.5498
Epoch 11: val_loss did not improve from 1.04920
1559/1559 [==============================] - 49s 31ms/step - loss: 1.0626 -
accuracy: 0.5498 - val_loss: 1.0516 - val_accuracy: 0.5540
Epoch 12/200
1557/1559 [============================>.] - ETA: 0s - loss: 1.0556 - accuracy:
0.5528
Epoch 12: val_loss improved from 1.04920 to 1.03518, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 49s 32ms/step - loss: 1.0556 -
accuracy: 0.5527 - val_loss: 1.0352 - val_accuracy: 0.5572
Epoch 13/200
1557/1559 [============================>.] - ETA: 0s - loss: 1.0514 - accuracy:
0.5555
Epoch 13: val_loss improved from 1.03518 to 1.03393, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 49s 32ms/step - loss: 1.0513 -
accuracy: 0.5555 - val_loss: 1.0339 - val_accuracy: 0.5632
Epoch 14/200
1559/1559 [==============================] - ETA: 0s - loss: 1.0465 - accuracy:
0.5576
Epoch 14: val_loss improved from 1.03393 to 1.02468, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 49s 31ms/step - loss: 1.0465 -
accuracy: 0.5576 - val_loss: 1.0247 - val_accuracy: 0.5670
Epoch 15/200
1559/1559 [==============================] - ETA: 0s - loss: 1.0430 - accuracy:
0.5599
Epoch 15: val_loss did not improve from 1.02468
1559/1559 [==============================] - 49s 31ms/step - loss: 1.0430 -
accuracy: 0.5599 - val_loss: 1.0314 - val_accuracy: 0.5635
Epoch 16/200
1559/1559 [==============================] - ETA: 0s - loss: 1.0397 - accuracy:
0.5606
Epoch 16: val_loss improved from 1.02468 to 1.02277, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 49s 32ms/step - loss: 1.0397 -
accuracy: 0.5606 - val_loss: 1.0228 - val_accuracy: 0.5654
Epoch 17/200
1559/1559 [==============================] - ETA: 0s - loss: 1.0365 - accuracy:
0.5628
Epoch 17: val_loss improved from 1.02277 to 1.02065, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 49s 32ms/step - loss: 1.0365 -
accuracy: 0.5628 - val_loss: 1.0206 - val_accuracy: 0.5666
Epoch 18/200
```

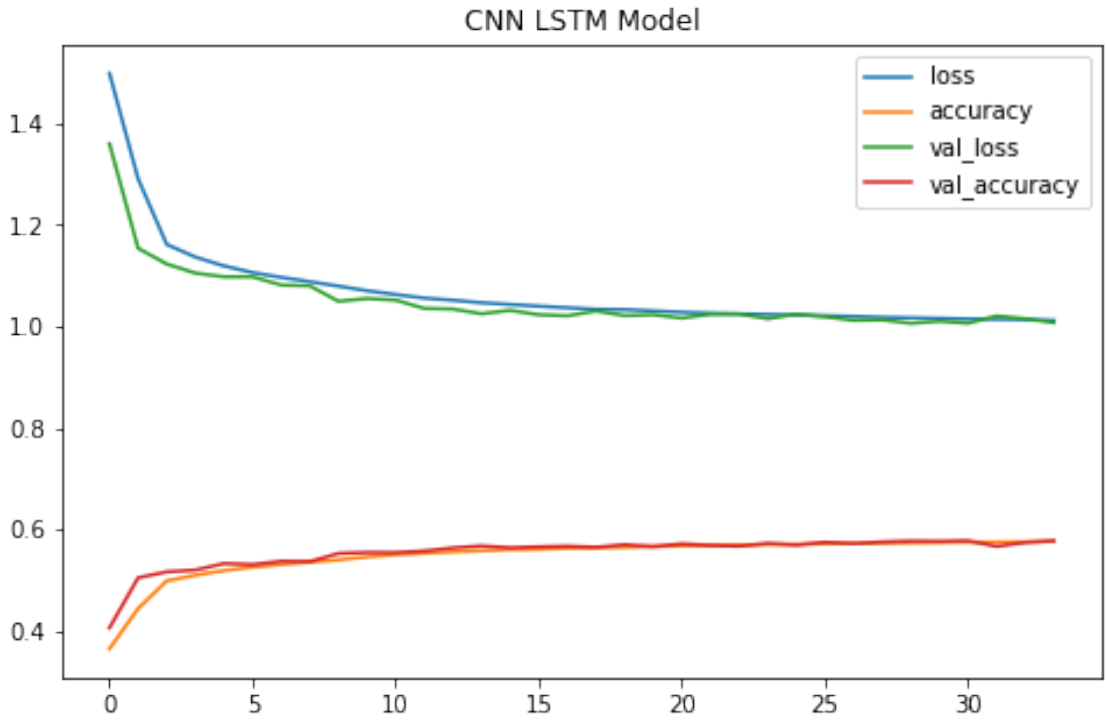
```
1559/1559 [==============================] - ETA: 0s - loss: 1.0335 - accuracy:
0.5637
Epoch 18: val_loss did not improve from 1.02065
1559/1559 [==============================] - 49s 31ms/step - loss: 1.0335 -
accuracy: 0.5637 - val_loss: 1.0301 - val_accuracy: 0.5644
Epoch 19/200
1559/1559 [==============================] - ETA: 0s - loss: 1.0330 - accuracy:
0.5644
Epoch 19: val_loss did not improve from 1.02065
1559/1559 [==============================] - 49s 31ms/step - loss: 1.0330 -
accuracy: 0.5644 - val_loss: 1.0211 - val_accuracy: 0.5696
Epoch 20/200
1559/1559 [==============================] - ETA: 0s - loss: 1.0302 - accuracy:
0.5662
Epoch 20: val_loss did not improve from 1.02065
1559/1559 [==============================] - 49s 31ms/step - loss: 1.0302 -
accuracy: 0.5662 - val_loss: 1.0227 - val_accuracy: 0.5656
Epoch 21/200
1559/1559 [==============================] - ETA: 0s - loss: 1.0276 - accuracy:
0.5671
Epoch 21: val_loss improved from 1.02065 to 1.01627, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 49s 32ms/step - loss: 1.0276 -
accuracy: 0.5671 - val_loss: 1.0163 - val_accuracy: 0.5714
Epoch 22/200
1559/1559 [==============================] - ETA: 0s - loss: 1.0259 - accuracy:
0.5685
Epoch 22: val_loss did not improve from 1.01627
1559/1559 [==============================] - 49s 31ms/step - loss: 1.0259 -
accuracy: 0.5685 - val_loss: 1.0237 - val_accuracy: 0.5681
Epoch 23/200
1559/1559 [==============================] - ETA: 0s - loss: 1.0244 - accuracy:
0.5687
Epoch 23: val_loss did not improve from 1.01627
1559/1559 [==============================] - 49s 31ms/step - loss: 1.0244 -
accuracy: 0.5687 - val_loss: 1.0242 - val_accuracy: 0.5667
Epoch 24/200
1559/1559 [==============================] - ETA: 0s - loss: 1.0233 - accuracy:
0.5699
Epoch 24: val_loss improved from 1.01627 to 1.01497, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 49s 32ms/step - loss: 1.0233 -
accuracy: 0.5699 - val_loss: 1.0150 - val_accuracy: 0.5723
Epoch 25/200
1559/1559 [==============================] - ETA: 0s - loss: 1.0221 - accuracy:
0.5704
Epoch 25: val_loss did not improve from 1.01497
1559/1559 [==============================] - 49s 31ms/step - loss: 1.0221 -
```

```
accuracy: 0.5704 - val_loss: 1.0238 - val_accuracy: 0.5688
Epoch 26/200
1559/1559 [==============================] - ETA: 0s - loss: 1.0208 - accuracy:
0.5713
Epoch 26: val_loss did not improve from 1.01497
1559/1559 [==============================] - 49s 31ms/step - loss: 1.0208 -
accuracy: 0.5713 - val_loss: 1.0187 - val_accuracy: 0.5744
Epoch 27/200
1559/1559 [==============================] - ETA: 0s - loss: 1.0196 - accuracy:
0.5717
Epoch 27: val_loss improved from 1.01497 to 1.01177, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 49s 32ms/step - loss: 1.0196 -
accuracy: 0.5717 - val_loss: 1.0118 - val_accuracy: 0.5719
Epoch 28/200
1559/1559 [==============================] - ETA: 0s - loss: 1.0177 - accuracy:
0.5724
Epoch 28: val_loss did not improve from 1.01177
1559/1559 [==============================] - 49s 31ms/step - loss: 1.0177 -
accuracy: 0.5724 - val_loss: 1.0127 - val_accuracy: 0.5751
Epoch 29/200
1559/1559 [==============================] - ETA: 0s - loss: 1.0166 - accuracy:
0.5731
Epoch 29: val_loss improved from 1.01177 to 1.00585, saving model to
saved_models/cnn_lstm_classification.hdf5
1559/1559 [==============================] - 49s 32ms/step - loss: 1.0166 -
accuracy: 0.5731 - val_loss: 1.0059 - val_accuracy: 0.5767
Epoch 30/200
1559/1559 [==============================] - ETA: 0s - loss: 1.0157 - accuracy:
0.5738
Epoch 30: val_loss did not improve from 1.00585
1559/1559 [==============================] - 49s 31ms/step - loss: 1.0157 -
accuracy: 0.5738 - val_loss: 1.0100 - val_accuracy: 0.5762
Epoch 31/200
1559/1559 [==============================] - ETA: 0s - loss: 1.0145 - accuracy:
0.5747
Epoch 31: val_loss did not improve from 1.00585
1559/1559 [==============================] - 49s 31ms/step - loss: 1.0145 -
accuracy: 0.5747 - val_loss: 1.0060 - val_accuracy: 0.5772
Epoch 32/200
1559/1559 [==============================] - ETA: 0s - loss: 1.0139 - accuracy:
0.5748
Epoch 32: val_loss did not improve from 1.00585
1559/1559 [==============================] - 49s 31ms/step - loss: 1.0139 -
accuracy: 0.5748 - val_loss: 1.0204 - val_accuracy: 0.5662
Epoch 33/200
1559/1559 [==============================] - ETA: 0s - loss: 1.0134 - accuracy:
0.5754
```

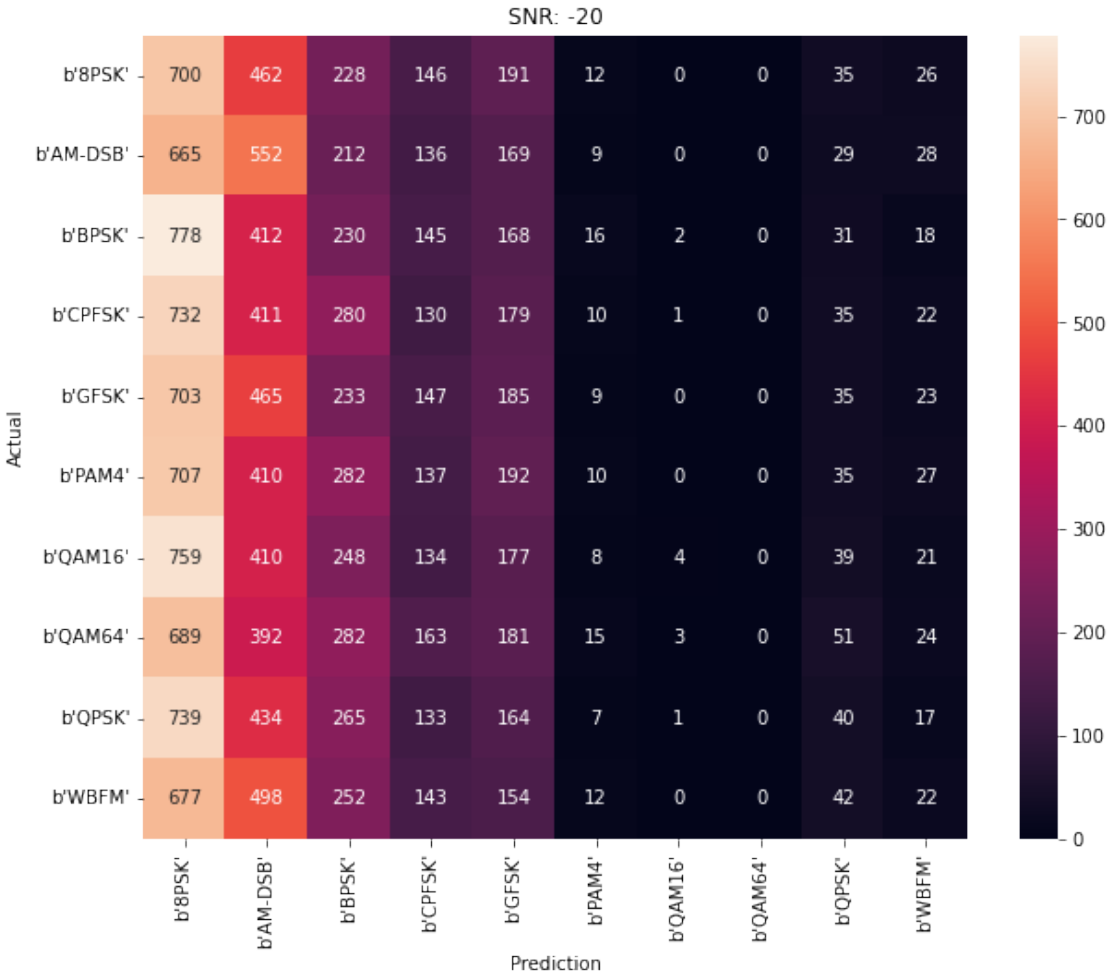
```
Epoch 33: val_loss did not improve from 1.00585
1559/1559 [==============================] - 49s 31ms/step - loss: 1.0134 -
accuracy: 0.5754 - val_loss: 1.0149 - val_accuracy: 0.5738
Epoch 34/200
1559/1559 [==============================] - ETA: 0s - loss: 1.0116 - accuracy:
0.5759
Epoch 34: val_loss did not improve from 1.00585
1559/1559 [==============================] - 49s 31ms/step - loss: 1.0116 -
accuracy: 0.5759 - val_loss: 1.0073 - val_accuracy: 0.5772
```

[47]:
```
clear_cache()
```

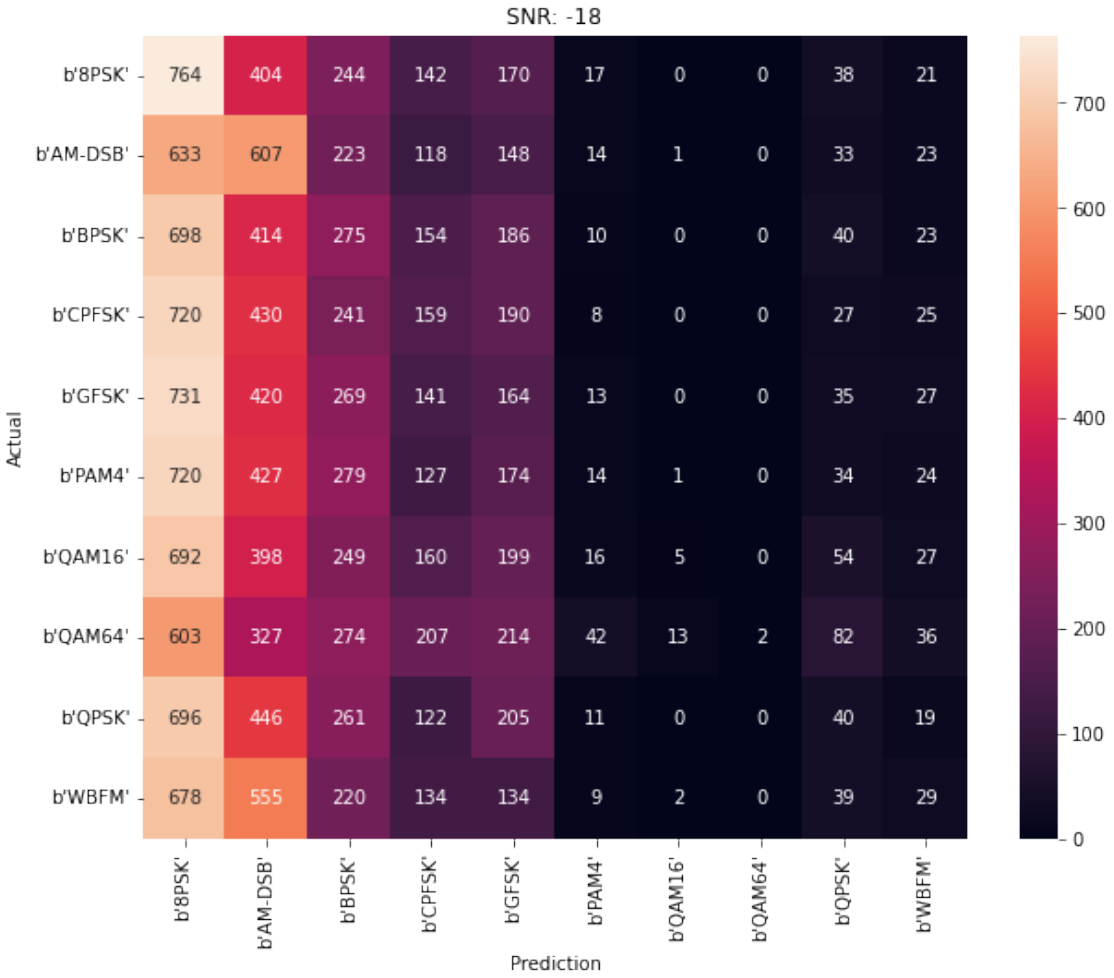
[53]:
```
plot_model_history(history, 'CNN LSTM Model')
model_scoring(cnn_lstm_model_2, X_test, testing_pair_labels)
```



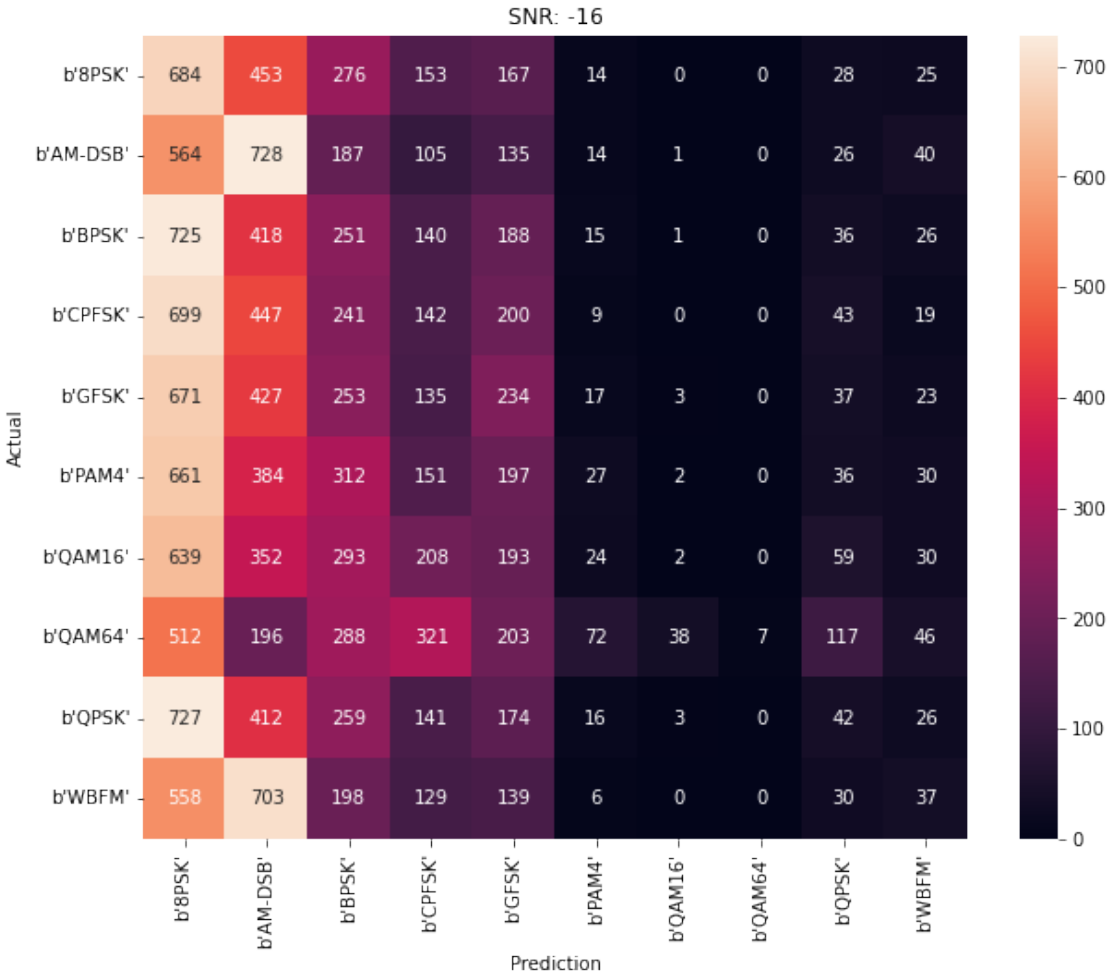

```
Accuracy at SNR = -20 is 0.10405555555555555%
```



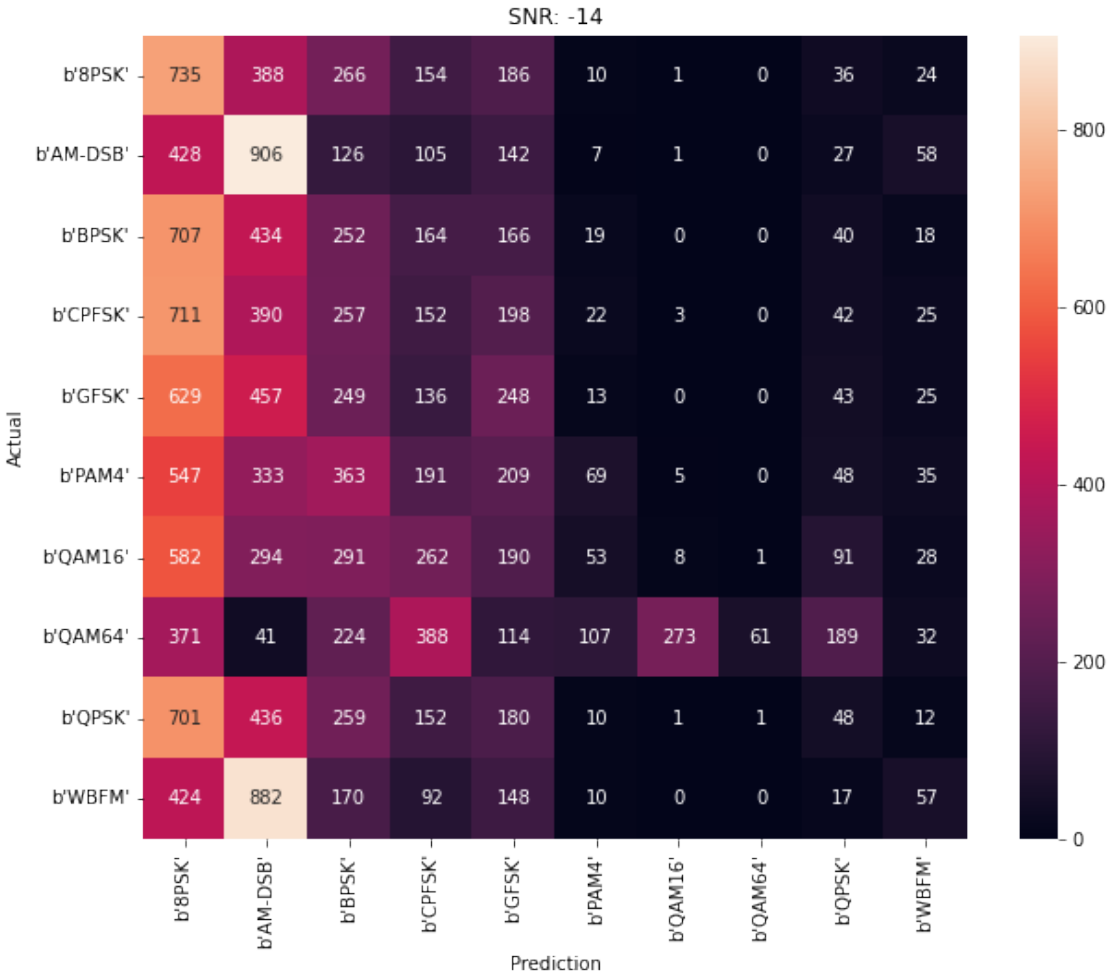

```
Accuracy at SNR = -18 is 0.11438888888888889%
```



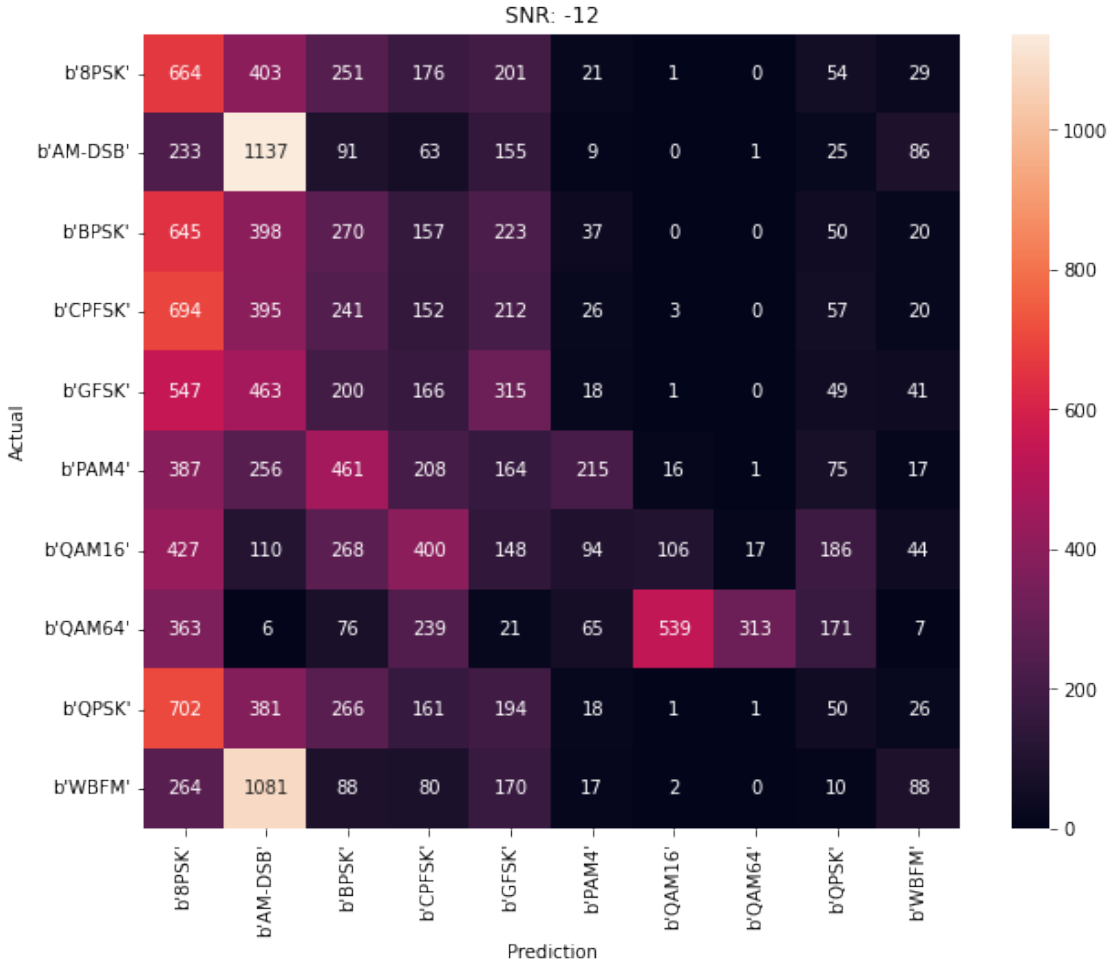

```
Accuracy at SNR = -16 is 0.11966666666666667%
```



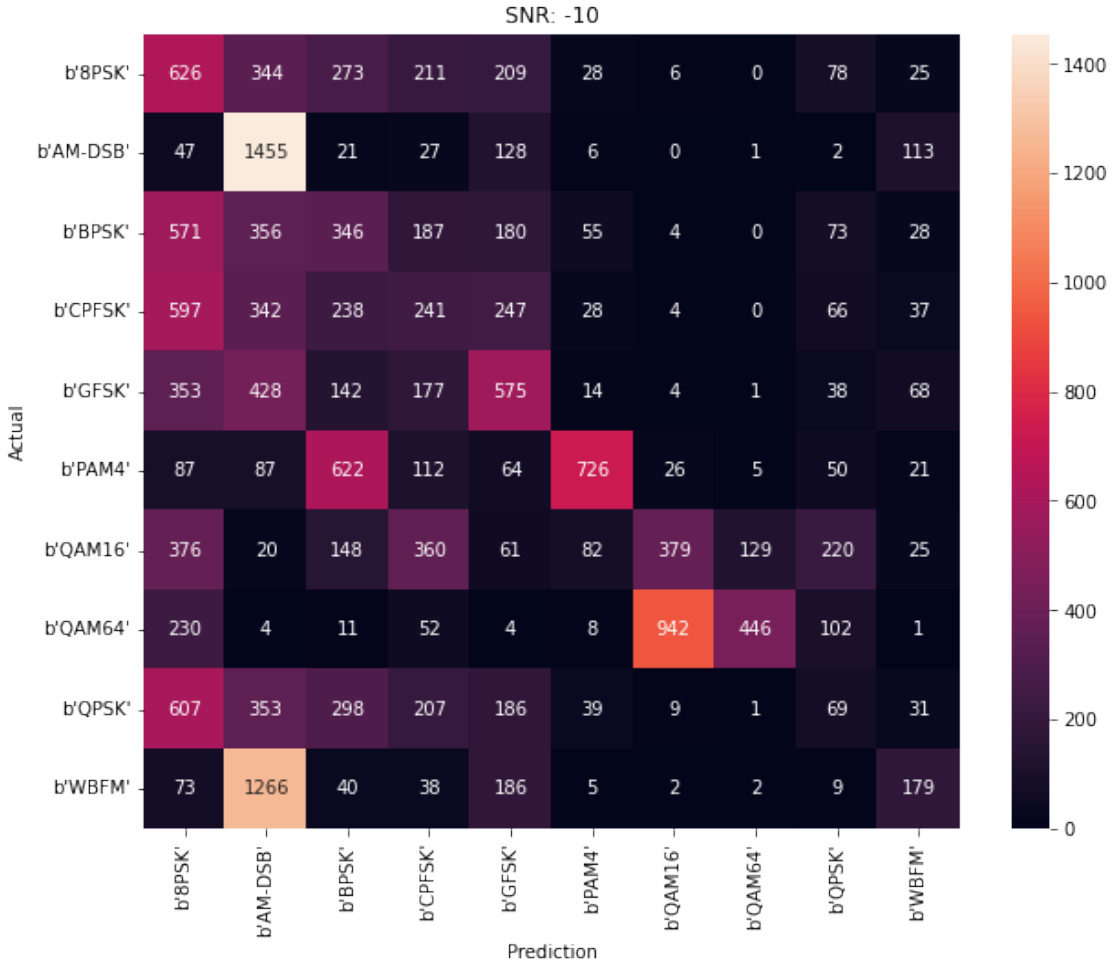

```
Accuracy at SNR = -14 is 0.1408888888888889%
```



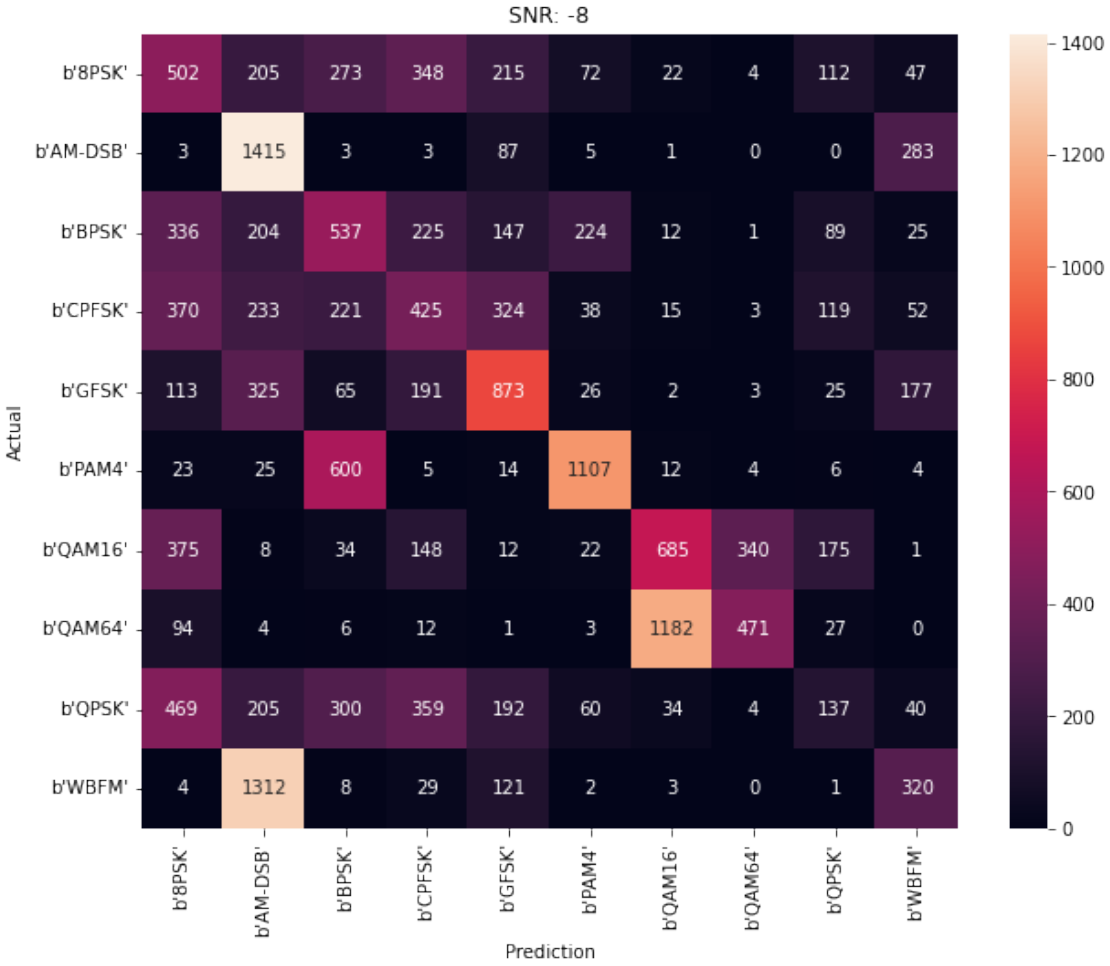

```
Accuracy at SNR = -12 is 0.18388888888888888%
```



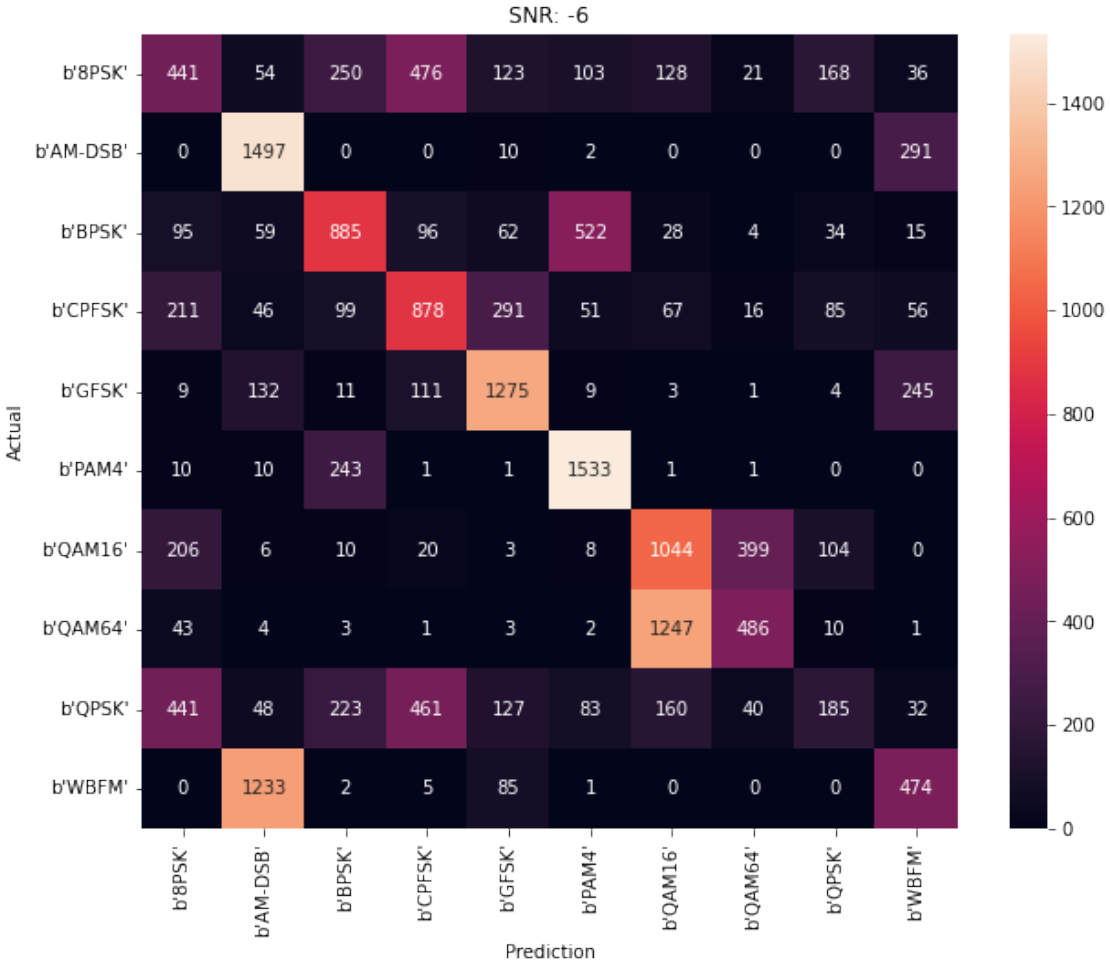

```
Accuracy at SNR = -10 is 0.2801111111111111%
```



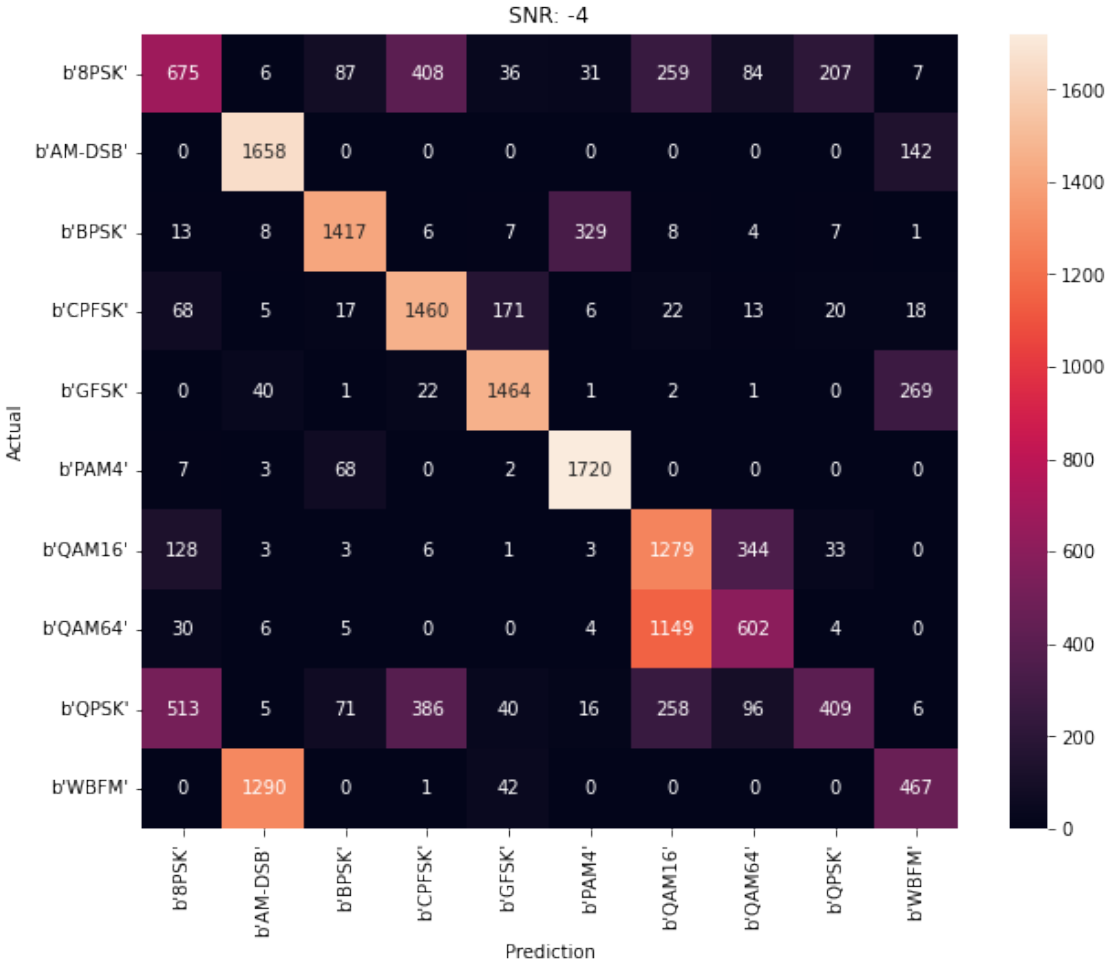

```
Accuracy at SNR = -8 is 0.3595555555555556%
```



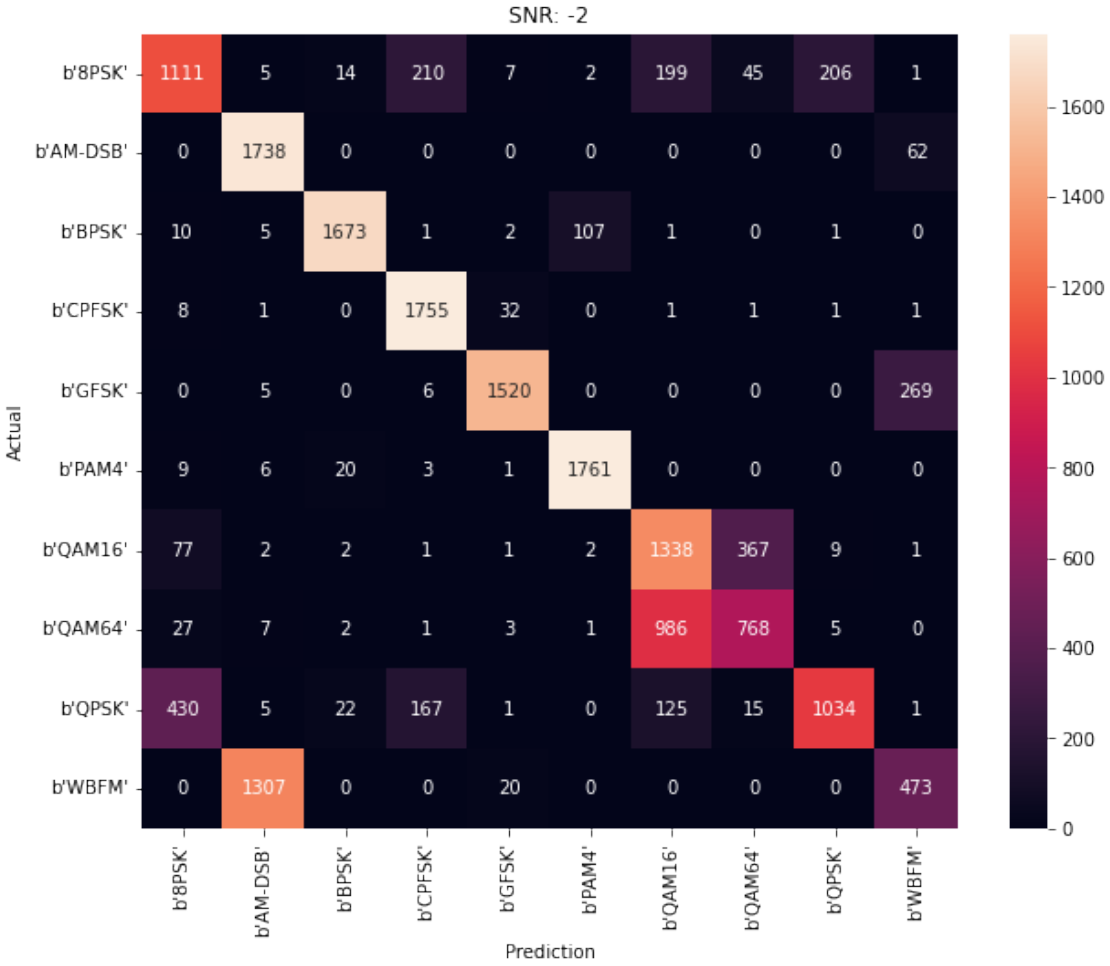

```
Accuracy at SNR = -6 is 0.4832222222222222%
```



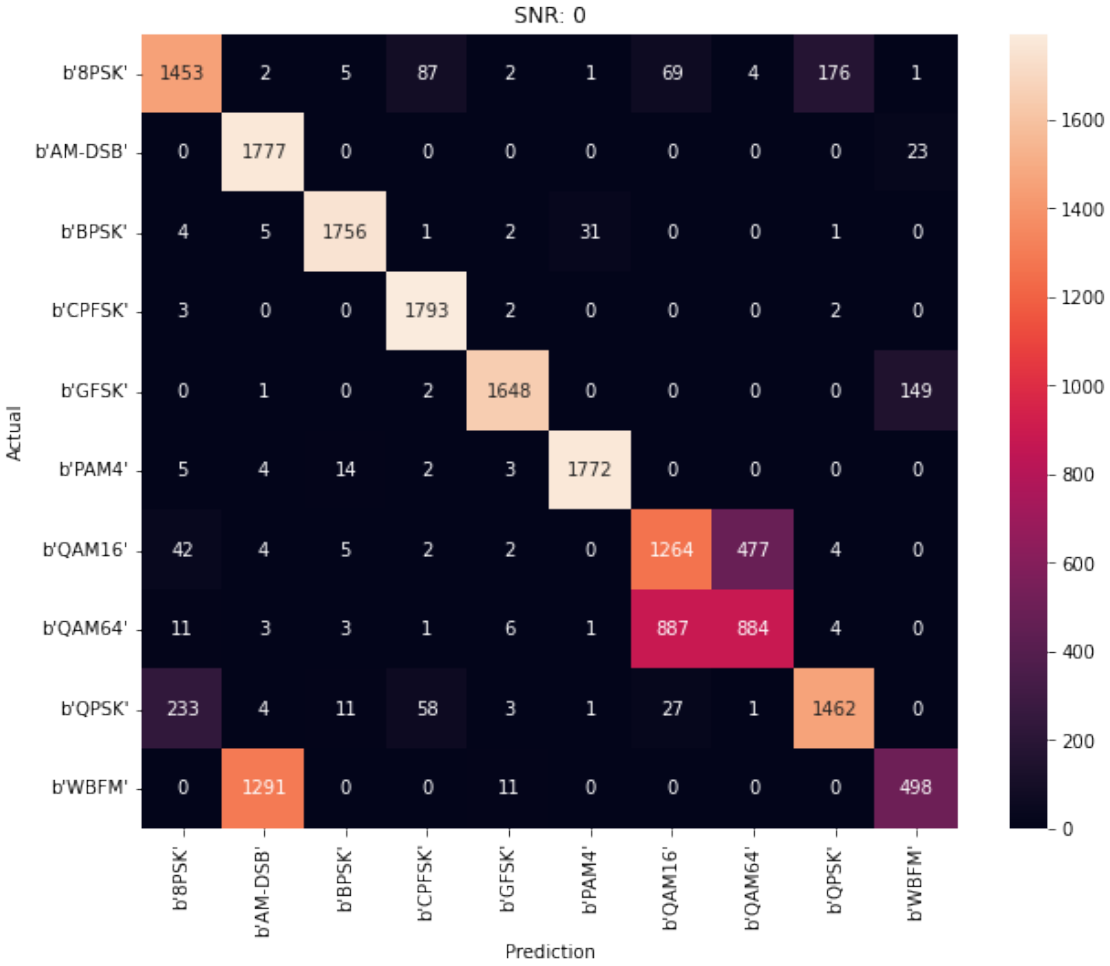

```
Accuracy at SNR = -4 is 0.6195%
```



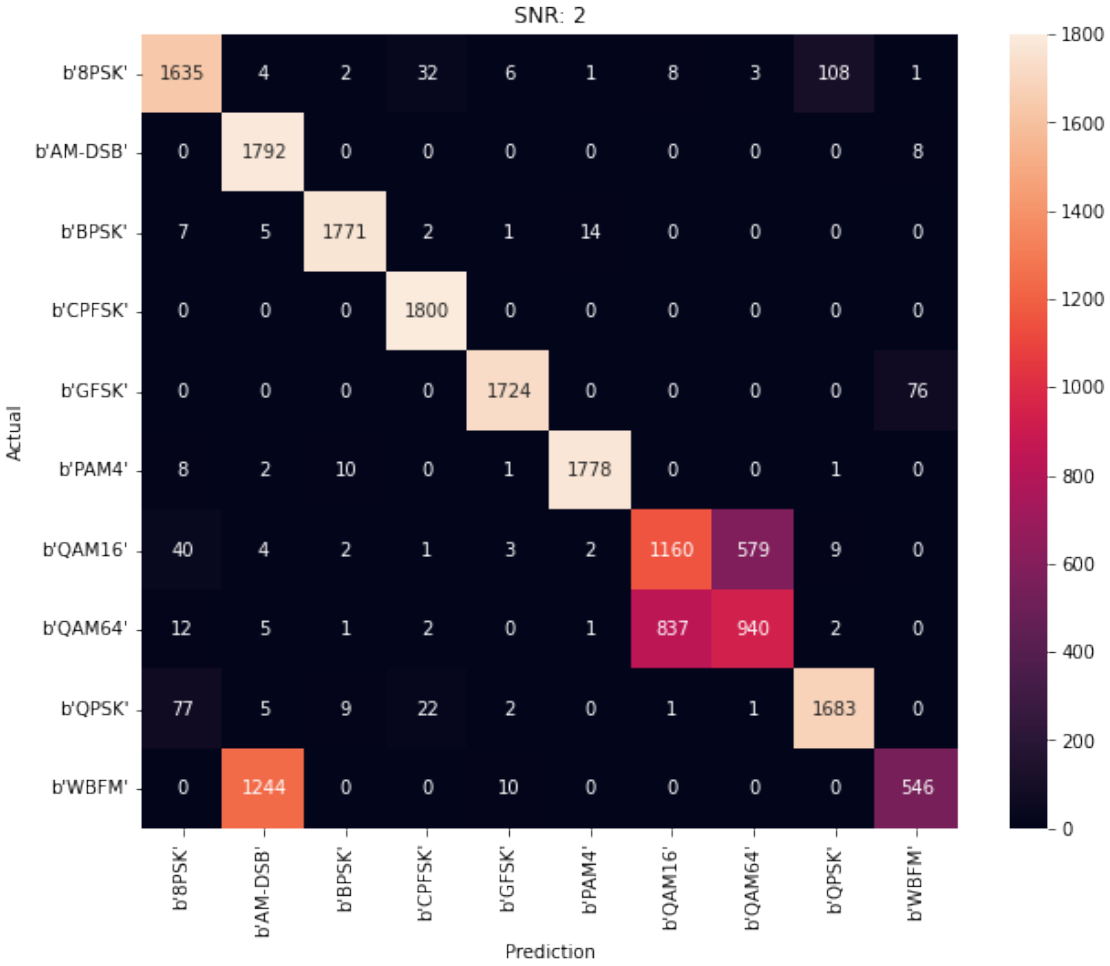

```
Accuracy at SNR = -2 is 0.7317222222222223%
```



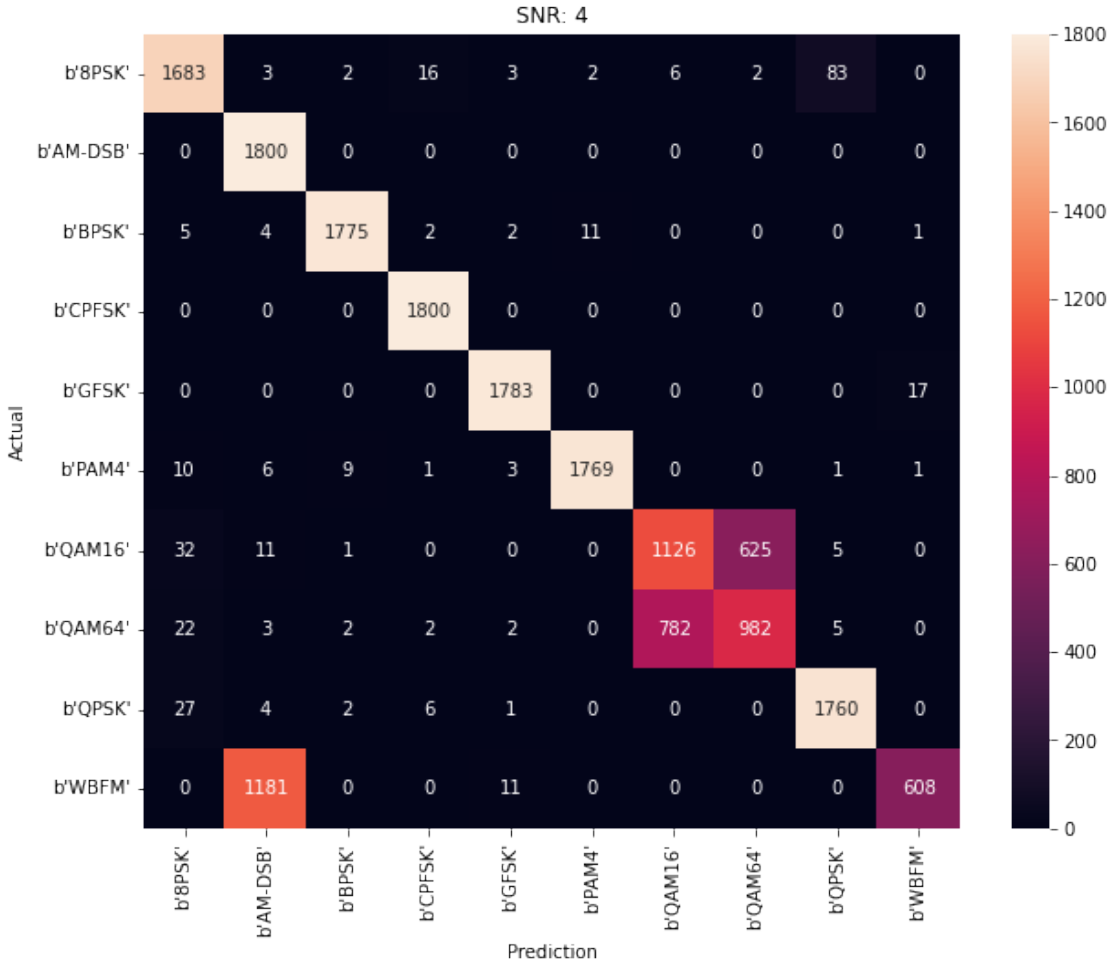

```
Accuracy at SNR = 0 is 0.7948333333333333%
```



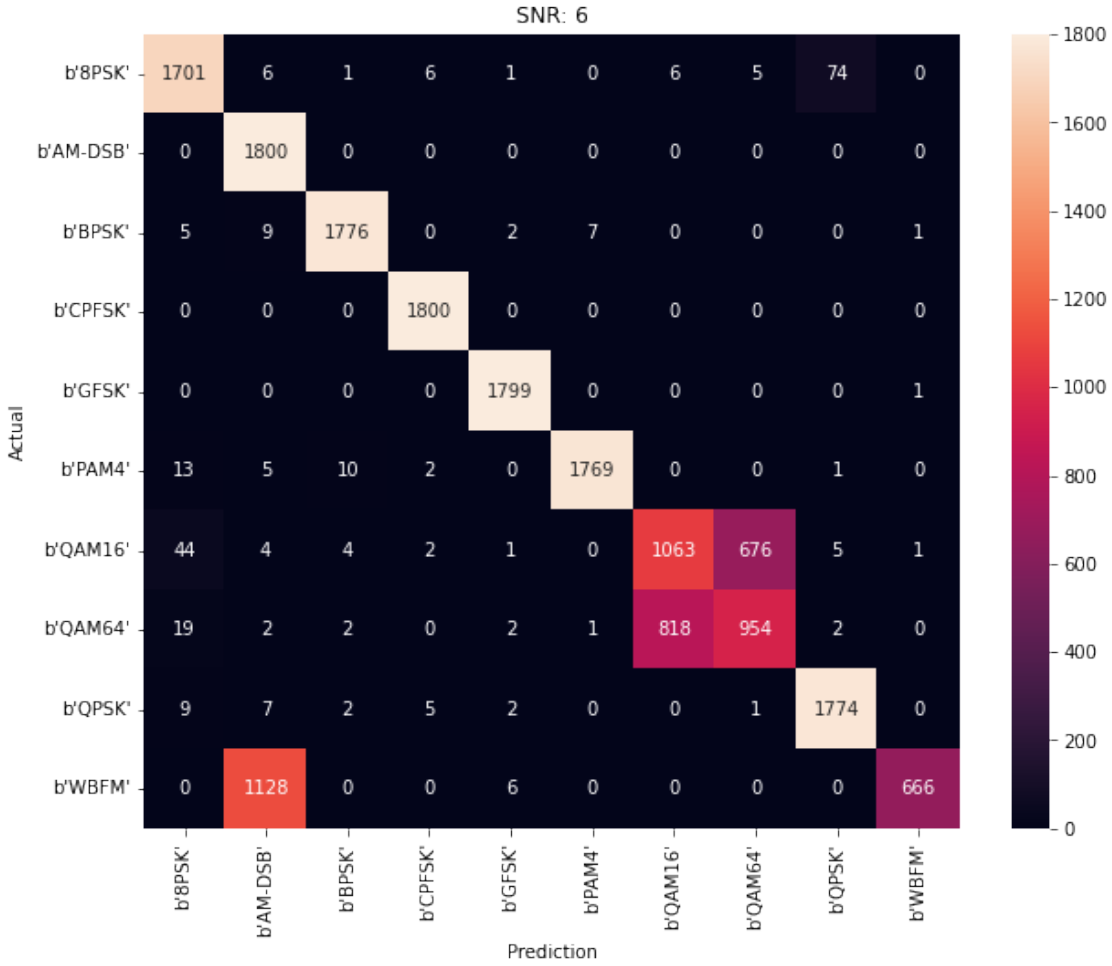

```
Accuracy at SNR = 2 is 0.8238333333333333%
```



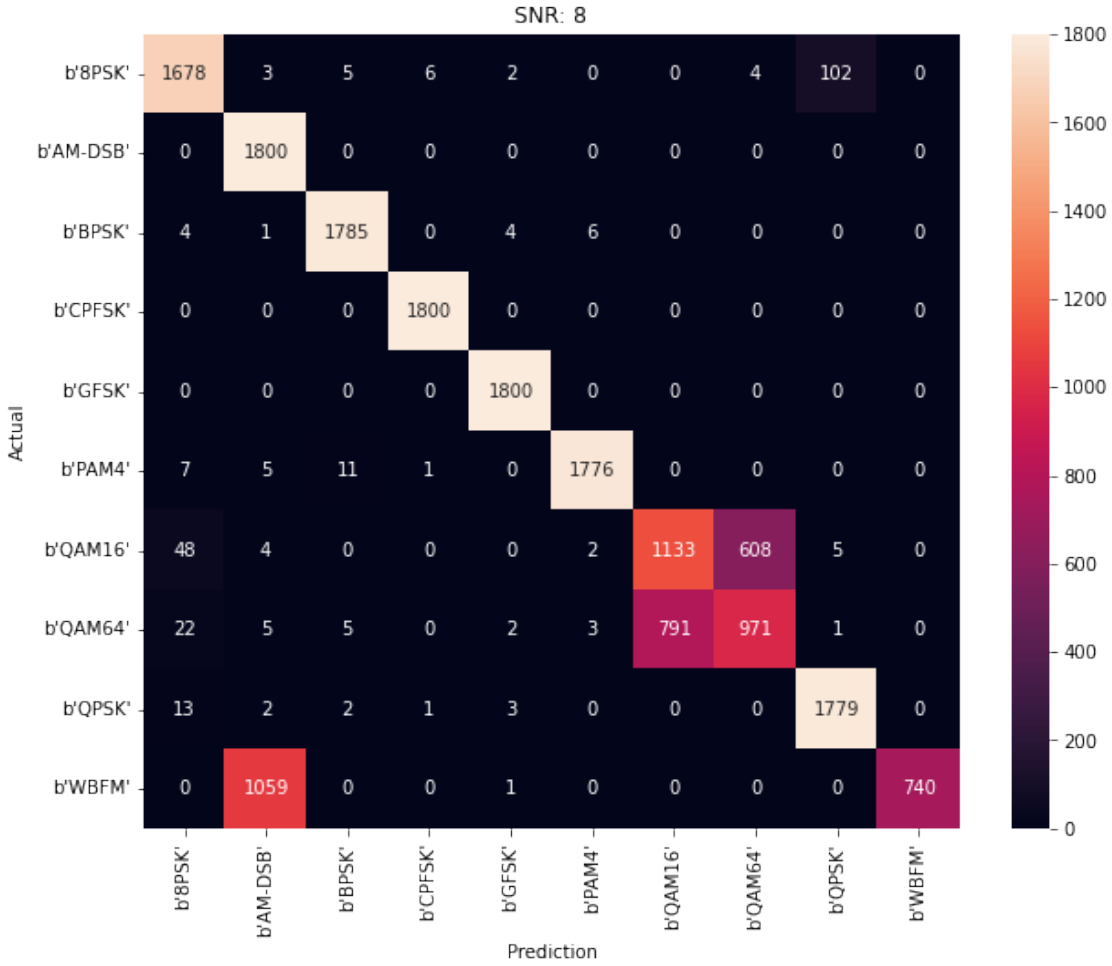
Accuracy at SNR = 4 is 0.8381111111111111%



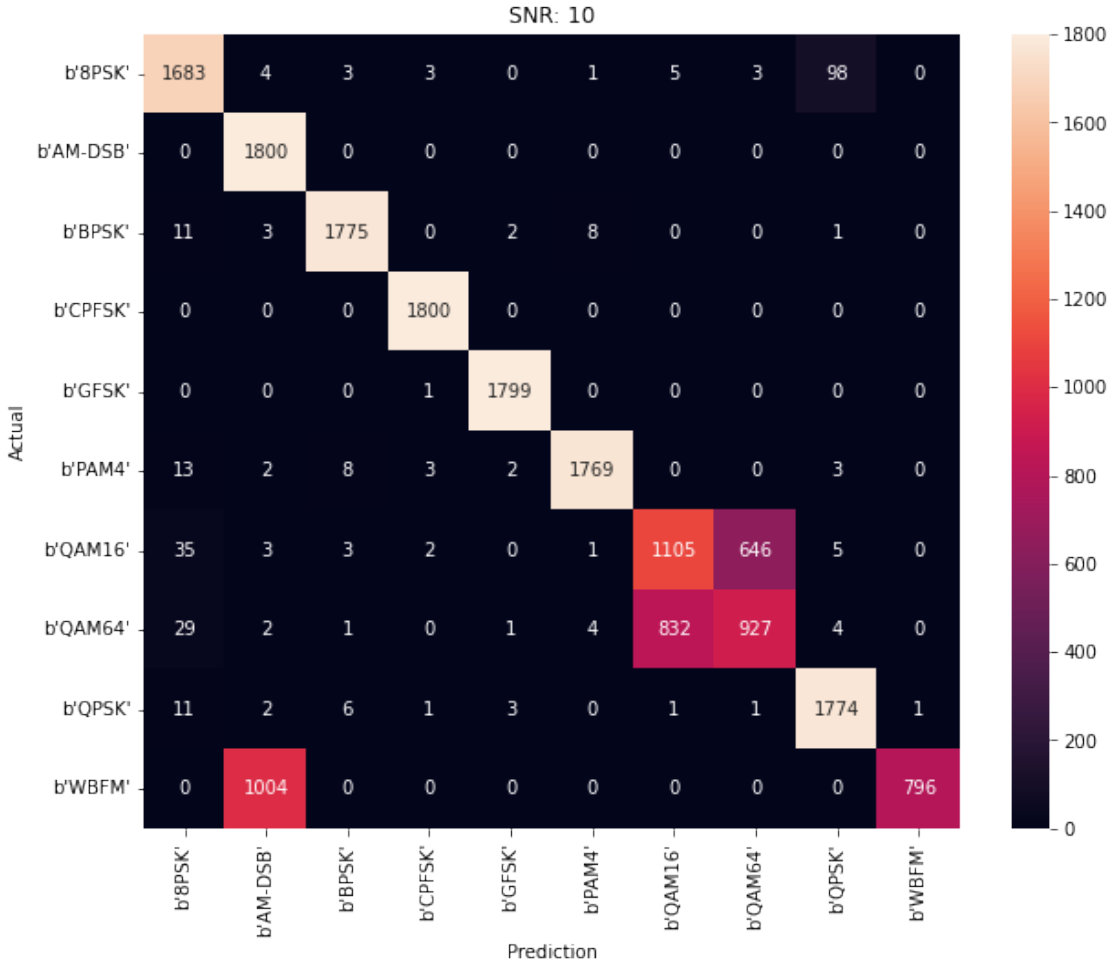
Accuracy at SNR = 6 is 0.839%



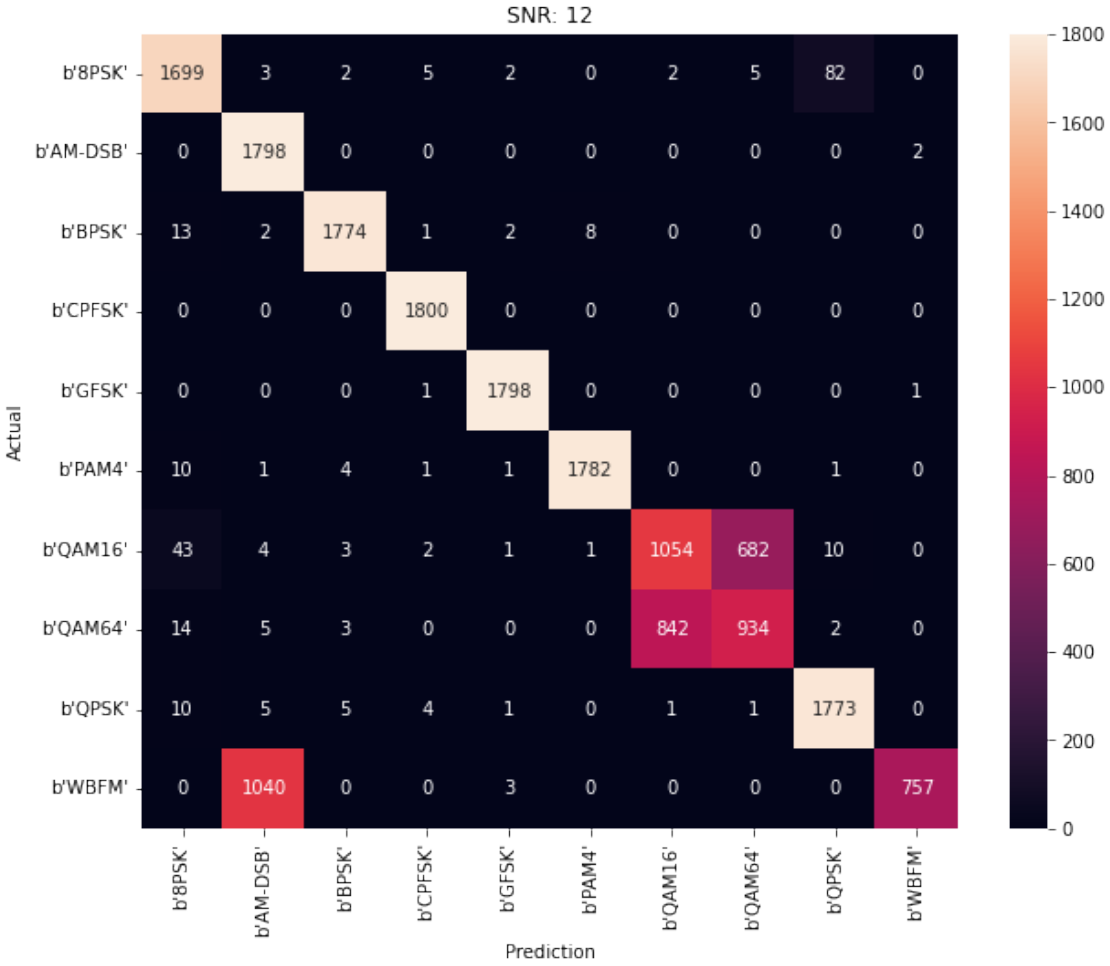

```
Accuracy at SNR = 8 is 0.8478888888888889%
```



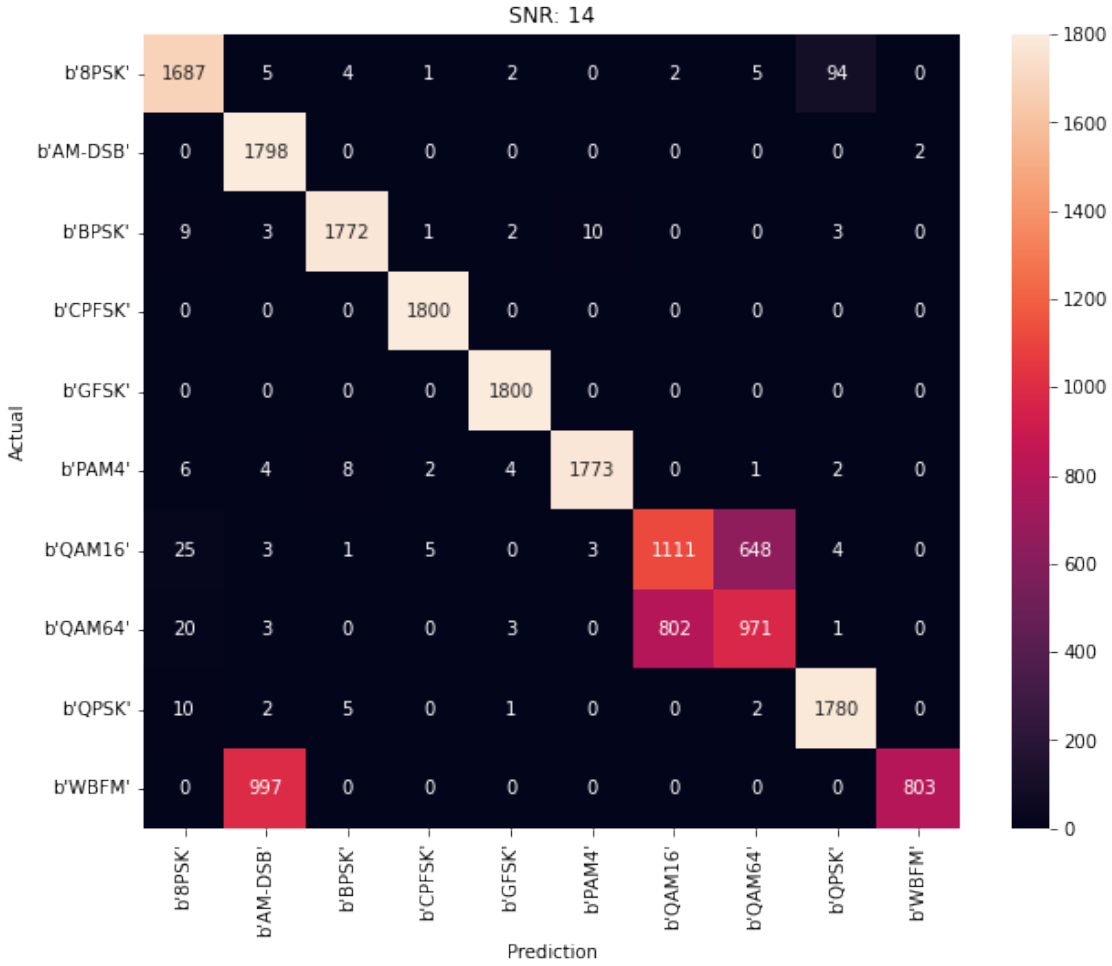

```
Accuracy at SNR = 10 is 0.846%
```



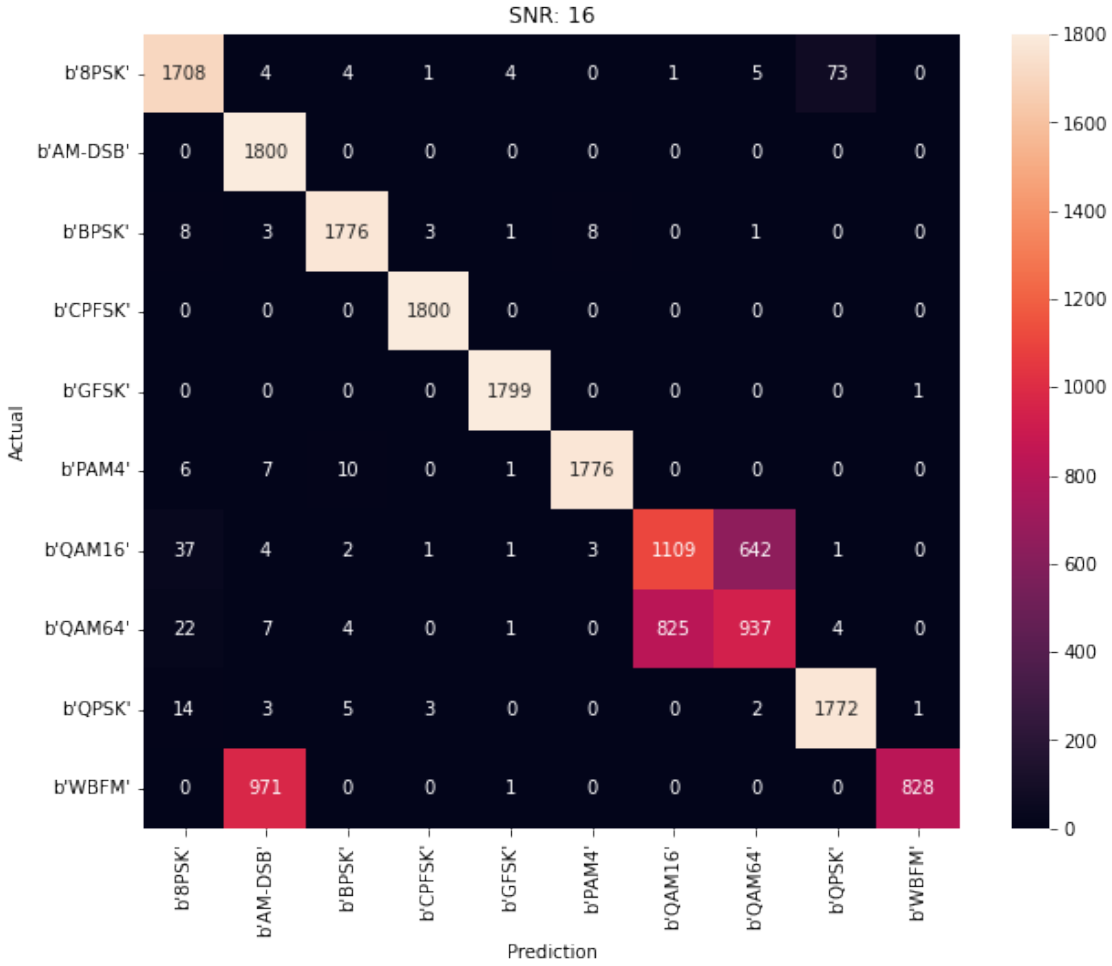
Accuracy at SNR = 12 is 0.8427222222222223%



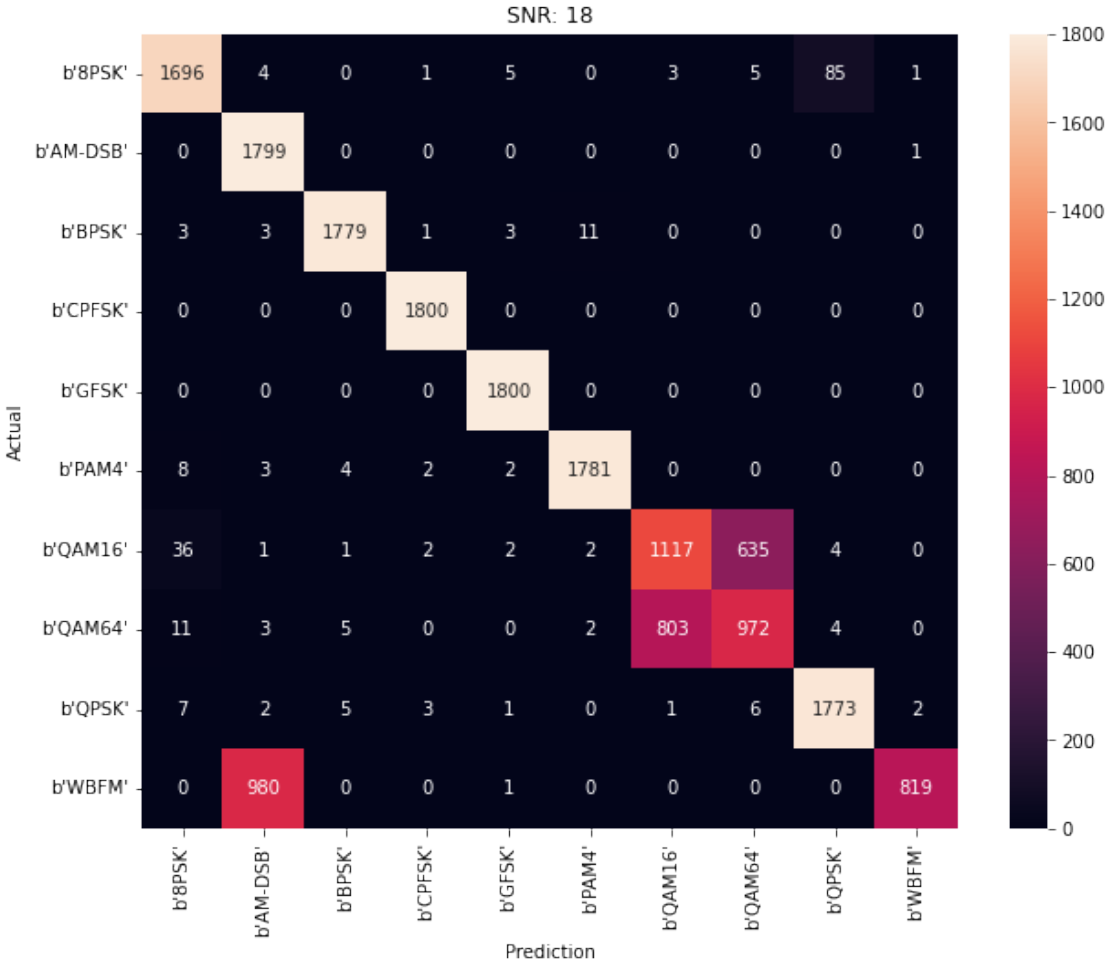

```
Accuracy at SNR = 14 is 0.8497222222222223%
```

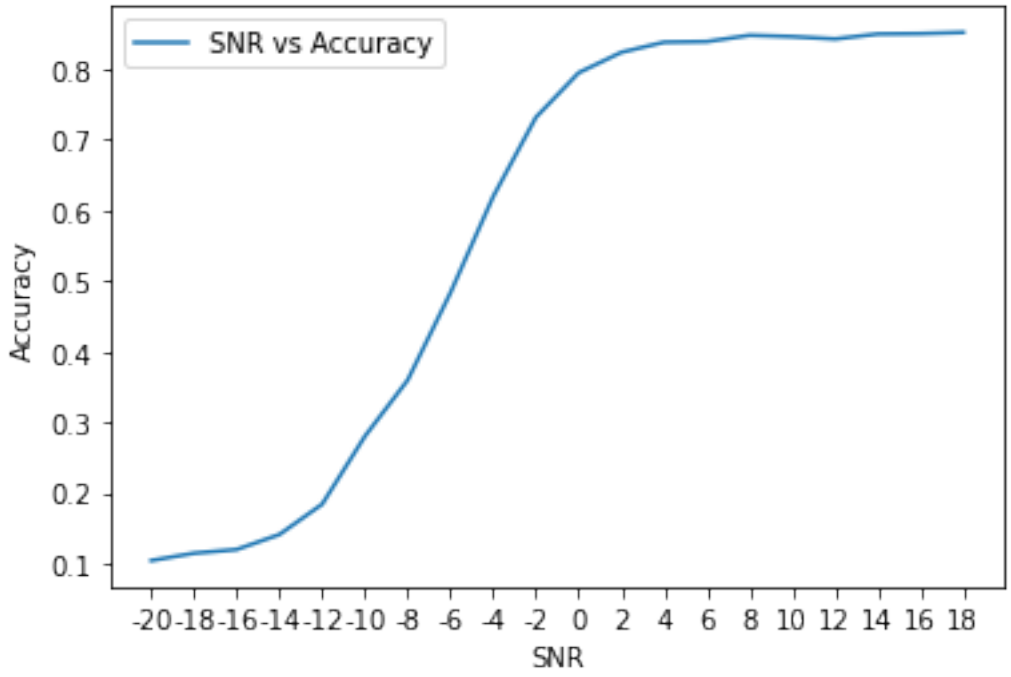



```
Accuracy at SNR = 16 is 0.8502777777777778%
```




```
Accuracy at SNR = 18 is 0.852%
```


SNR: 18
b'8PSK' 1696 4 0 1 5 0 3 5 85 1
b'AM-DSB' 0 1799 0 0 0 0 0 0 0 1
b'BPSK' 3 3 1779 1 3 11 0 0 0 0
b'CPFSK' 0 0 0 1800 0 0 0 0 0 0
b'GFSK' 0 0 0 0 1800 0 0 0 0 0
b'PAM4' 8 3 4 2 2 1781 0 0 0 0
b'QAM16' 36 1 1 2 2 2 1117 635 4 0
b'QAM64' 11 3 5 0 0 2 803 972 4 0
b'QPSK' 7 2 5 3 1 0 1 6 1773 2
b'WBFM' 0 980 0 0 1 0 0 0 0 819
Actual
b'8PSK' b'AM-DSB' b'BPSK' b'CPFSK' b'GFSK' b'PAM4' b'QAM16' b'QAM64' b'QPSK' b'WBFM'
Prediction
1800 1600 1400 1200 1000 800 600 400 200 0


SNR vs Accuracy
Accuracy
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
-20 -18 -16 -14 -12 -10 -8 -6 -4 -2 0 2 4 6 8 10 12 14 16 18
SNR

[ ]: