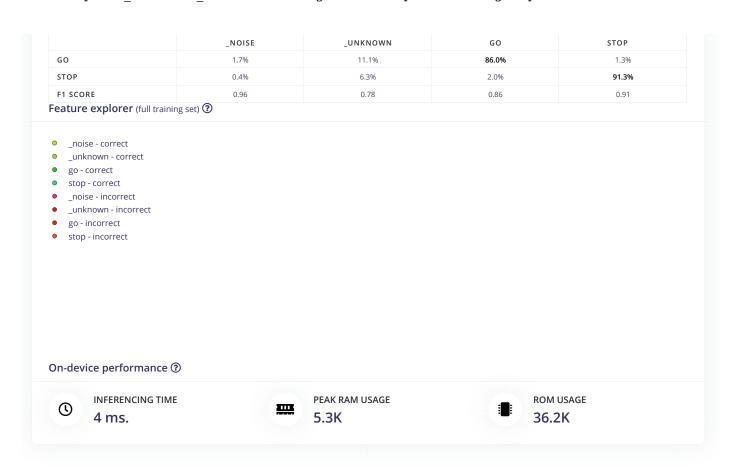
Click to set a description for this version <u>#1</u> **Neural Network settings** : **Training settings** Minimum confidence rating ? 0.60 Neural network architecture import tensorflow as tf from tensorflow.keras.models import Sequential from tensorflow.keras.layers import Dense, InputLayer, Dropout, Conv1D,
 Conv2D, Flatten, Reshape, MaxPooling1D, MaxPooling2D, BatchNormalization from tensorflow.keras.optimizers import Adam sys.path.append('./resources/libraries') 6 import ei_tensorflow.training # model architecture model = Sequential() model.add(Reshape((int(input_length / 13), 13), input_shape=(input_length model.add(Conv1D(8, kernel_size=3, activation='relu', padding='same'))
model.add(MaxPooling1D(pool_size=2, strides=2, padding='same')) 11 12 model.add(Dropout(0.25)) model.add(Conv1D(16, kernel_size=3, activation='relu', padding='same')) model.add(MaxPooling1D(pool_size=2, strides=2, padding='same'))
model.add(Dropout(0.25))
model.add(Flatten()) 15 16 17 model.add(Dense(classes, activation='softmax', name='y_pred')) 19 # this controls the learning rate
opt = Adam(lr=0.005, beta_1=0.9, beta_2=0.999)
this controls the batch size, or you can manipulate the tf.data.Dataset 20 21 22 objects yourself BATCH_SIZE = 512 23 train_dataset, validation_dataset = ei_tensorflow.training.set_batch_size
 (BATCH_SIZE, train_dataset, validation_dataset)
callbacks.append(BatchLoggerCallback(BATCH_SIZE, train_sample_count)) 24 25 # train the neural network
model.compile(loss='categorical_crossentropy', optimizer=opt, metrics 28 Start training **Training output** Model Model version: 3 Quantized (int8) ▼ Last training performance (validation set) **ACCURACY** LOSS _~ % 87.8% 0.35 Confusion matrix (validation set) UNKNOWN NOISE 95.9% 1.6% 0% 2.4% Ø _UNKNOWN 3.1% 77.1% 12.3% 7.5%

1 of 2 4/17/21, 14:36



© 2020 EdgeImpulse Inc. All rights reserved

2 of 2 4/17/21, 14:36