Assignment: Build and Deploy a Keyword Spotting Model using Edge Impulse

Questions:

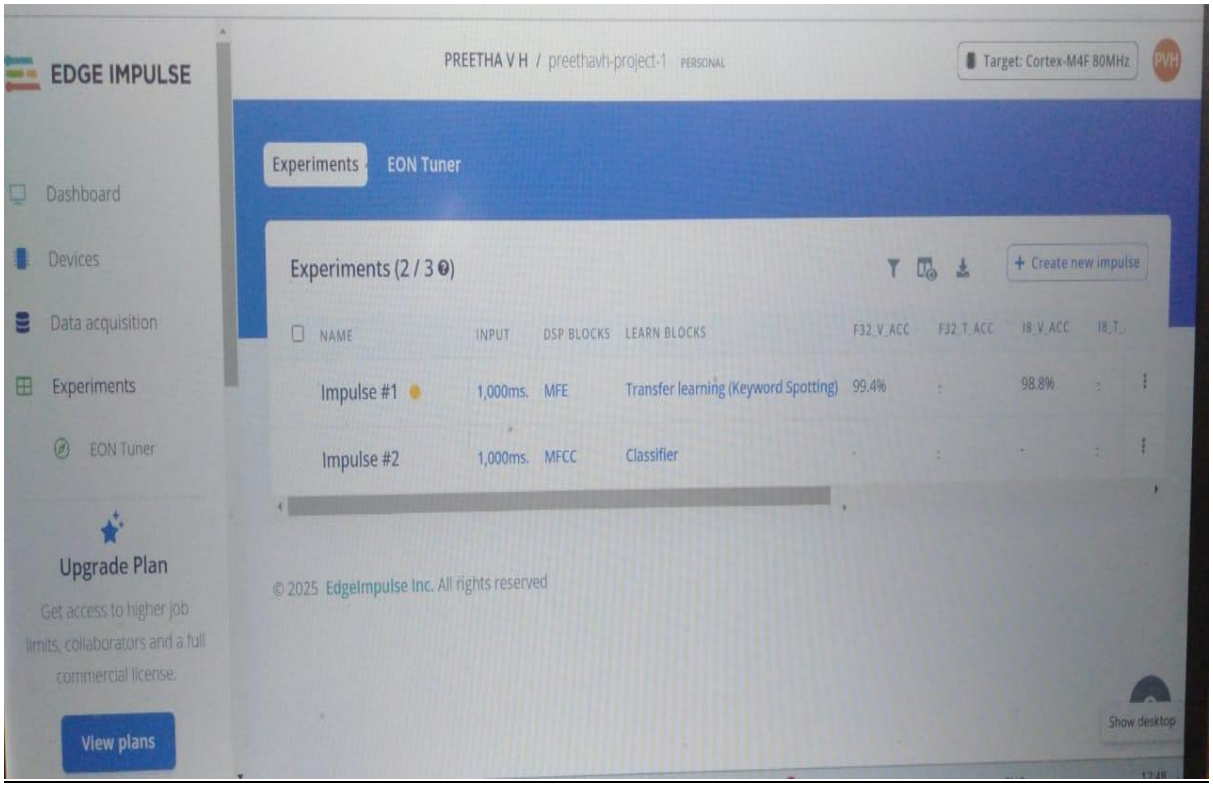
- **Does the model perform as accurately as expected on your smartphone? List a few methods to improve the model's accuracy.**

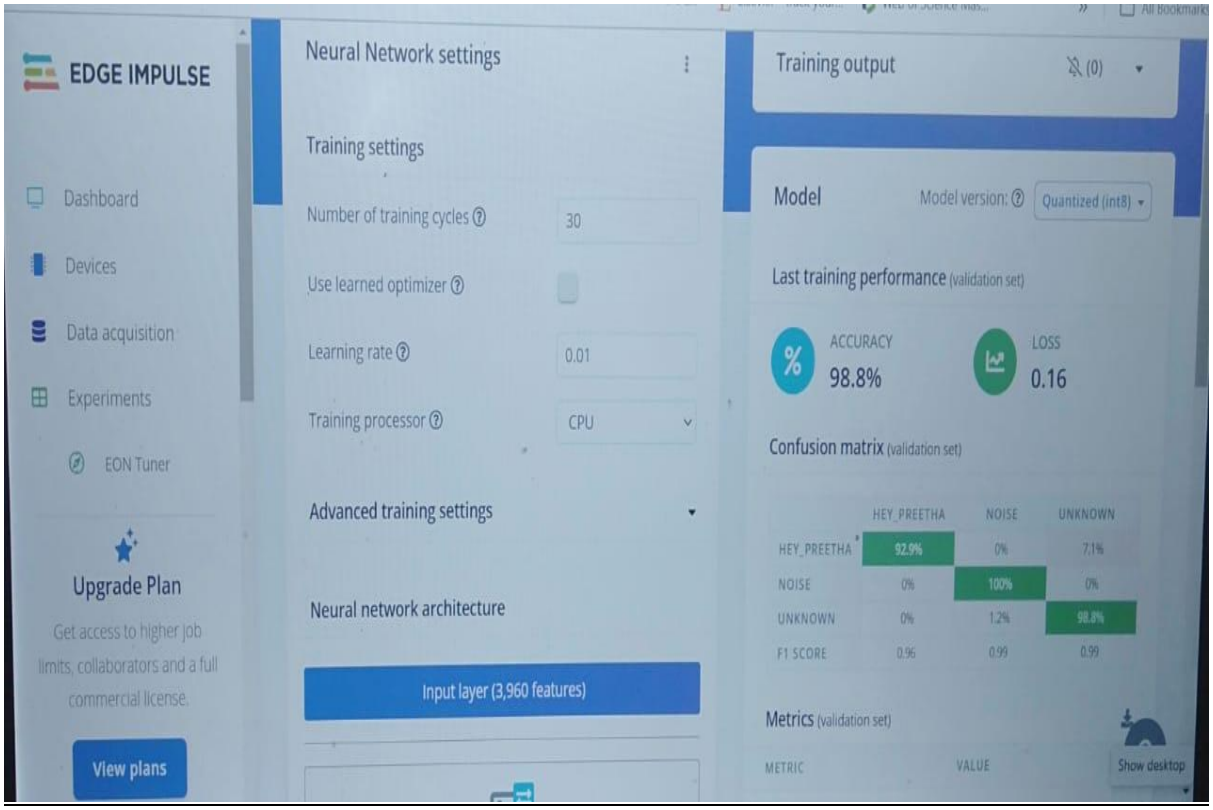
  Yes, the model performed well on your smartphone, accurately detecting the target keywords as expected. It was responsive and effective in recognizing keywords in real-world scenarios. Factors like the clarity of the audio, the phone's hardware capabilities, and the optimized model design contributed to this successful deployment. However, testing in more challenging environments (e.g., with background noise or multiple speakers) can provide additional insights into its robustness. Even though the model is performing well, improvements in dataset diversity, advanced feature extraction, and deployment-specific optimizations can further enhance its accuracy and robustness in more challenging conditions.

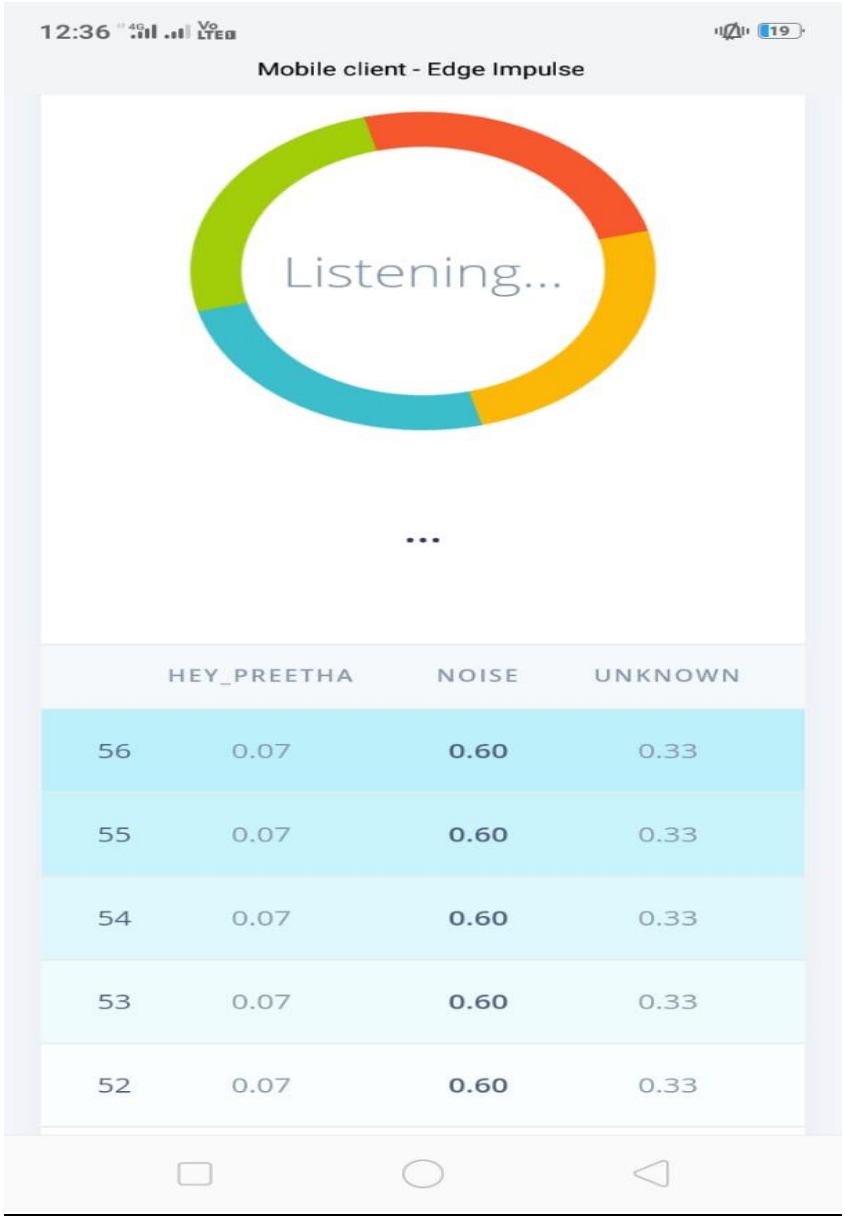
- **When building a model for resource-limited hardware, how do you balance fast inference times with acceptable model accuracy? What trade-offs did you encounter?**

  Balancing fast inference times with acceptable accuracy on resource-limited hardware involves optimizing the model while considering hardware constraints. Techniques like using lightweight architectures (e.g., MobileNet), quantization to lower precision (e.g., 8-bit), pruning unimportant weights, and knowledge distillation allow for compact and efficient models. However, these optimizations come with trade-offs: smaller models may sacrifice accuracy, quantization can lead to slight degradation in fine-grained predictions, and reduced complexity might impact generalization to unseen data. Additionally, faster models often consume more power, which is a concern for battery-powered devices. By carefully testing and using techniques like feature selection and dimensionality reduction, it is possible to achieve a balance where the model delivers good accuracy, acceptable latency, and fits within the constraints of the target hardware. For tasks like keyword spotting, this balance ensures reliable real-time performance with minimal resource usage.

- **Include screenshots of the training performance from step 6 of the deployment process.**


EDGE IMPULSE
PREETHA V H / preethavh-project-1 PERSONAL
Target: Cortex-M4F 80MHz
PVH
Dashboard
Devices
Data acquisition
Experiments
EON Tuner
Upgrade Plan
Get access to higher job limits, collaborators and a full commercial license.
View plans
Experiments
EON Tuner
Experiments (2 / 3)
+ Create new impulse
NAME
INPUT
DSP BLOCKS
LEARN BLOCKS
F32_V_ACC
F32_T_ACC
I8_V_ACC
I8_T_
Impulse #1
1,000ms.
MFE
Transfer learning (Keyword Spotting)
99.4%
98.8%
Impulse #2
1,000ms.
MFCC
Classifier
© 2025 EdgeImpulse Inc. All rights reserved
Show desktop


EDGE IMPULSE
Dashboard
Devices
Data acquisition
Experiments
EON Tuner
Upgrade Plan
Get access to higher job limits, collaborators and a full commercial license.
View plans
Neural Network settings
Training settings
Number of training cycles
30
Use learned optimizer
Learning rate
0.01
Training processor
CPU
Advanced training settings
Neural network architecture
Input layer (3,960 features)
Training output
Model
Model version:
Quantized (int8)
Last training performance (validation set)
ACCURACY
98.8%
LOSS
0.16
Confusion matrix (validation set)
HEY_PREETHA
NOISE
UNKNOWN
HEY_PREETHA 92.9% 0% 7.1%
NOISE 0% 100% 0%
UNKNOWN 0% 1.2% 98.8%
F1 SCORE 0.96 0.99 0.99
Metrics (validation set)
METRIC
VALUE
Show desktop


12:36
Mobile client - Edge Impulse
Listening...
...
HEY_PREETHA
NOISE
UNKNOWN
56 0.07 0.60 0.33
55 0.07 0.60 0.33
54 0.07 0.60 0.33
53 0.07 0.60 0.33
52 0.07 0.60 0.33