

## 2021 | Bose Labs Take-home iOS Coding Assignment

### Upcycled Baby Monitor

#### Introduction

Instead of throwing it out, do you have an old smartphone that you hope to repurpose to something useful? In this take-home assignment, you will create an iOS application that allows people to turn their old smartphones into baby monitors, potentially extending the lifetime of these devices and reducing E-waste. How wonderful!

To simplify the problem, you will only need to implement one feature — **use a deep learning model to detect the sound of a crying baby and display on the screen when such events happened**. The figure below provides an example user interface.



#### Setup

- You can find a XCode project we prepared in the zip file we sent you, this served as a boilerplate to simplify the assignment.
- We have already trained a deep-learning-based sound classification model using [Turicreate](#) and the [ESC-10 dataset](#). This model can detect 10 types of sound and one of them is the sound of crying baby. You may read the mlmodel file for more detail.
- This deep learning model depends on a class called TCSoundClassifierPreprocessing, which will take in 15600 audio samples at a time (sampled at 16000Hz) to produce the frequency-domain features used by the final VGG-ish deep learning model. You may read [here](#) for more detail. You don't need to worry too much about this preprocessing component, and we've bundled this class for you in the AudioPreprocessingLib folder.

- We've also written an AudioManager to help you grab microphone in real-time. You may find the documentation of [AVAudioEngine](#) and [AVAudioNode](#) to be useful.
- This boilerplate could also run in a Simulator, in case you do not have an iPhone.

### What you need to do

- You need to pipe the audio data into the deep learning model we prepared. You may find [SNAudioStreamAnalyzer](#) useful as it will make sample rate conversion and buffering easier. However, you are open to use any other tools you are more familiar with. Note that the microphone by default sampled at 44.1kHz while our model is trained for 16kHz.
- If you do use third party frameworks, please make sure you document how to setup them in the README.md.
- You will then parse the output of the deep learning model and filter on the confidence level of the prediction (set the threshold to be 50% in this assignment) as you don't want an event with low confidence to show up on the screen. Note that we only care about crying baby sound, not other types of sound.
- You will then need to create a table view that displays the detected crying baby events with their timestamps formatted in a user-friendly fashion.
- Since we don't want to clutter the screen with consecutive crying baby events that happened close to each other, you will need to merge predictions together. In this assignment, we'll define a crying baby session to be just 20 seconds long. That is, if you detect a crying baby sound, any other crying baby events happened within 20 seconds of the first event will be considered the same event. E.g., assume these are the timestamps of crying baby events: [0<sup>th</sup> second, 5, 19, 21, 41], after merging it will become [0, 21, 41]. You will only need to display the merged events on the screen.
- You may use either SwiftUI or UIKit for interface development.

### Deliverable

- Email us a zip file that contains your code and the necessary setup documentation in README.md if you use any additional frameworks.

### Evaluation

- You have 7 days to work on this assignment from the time we email it to you. From our records, the assignment could be completed in a much shorter time. We want to provide plenty of flexibility to you to accommodate your unique situation. If you complete it earlier, please submit it earlier.
- The goal of this assignment is to help us evaluate:
  - Your skill in iOS development
  - Your capability to find and read documentation to solve problems
  - Your understanding of machine learning models with regard to their input and output
  - Your understanding of basic audio signal processing

- We will evaluate your code based on its function and code quality (including code organization).