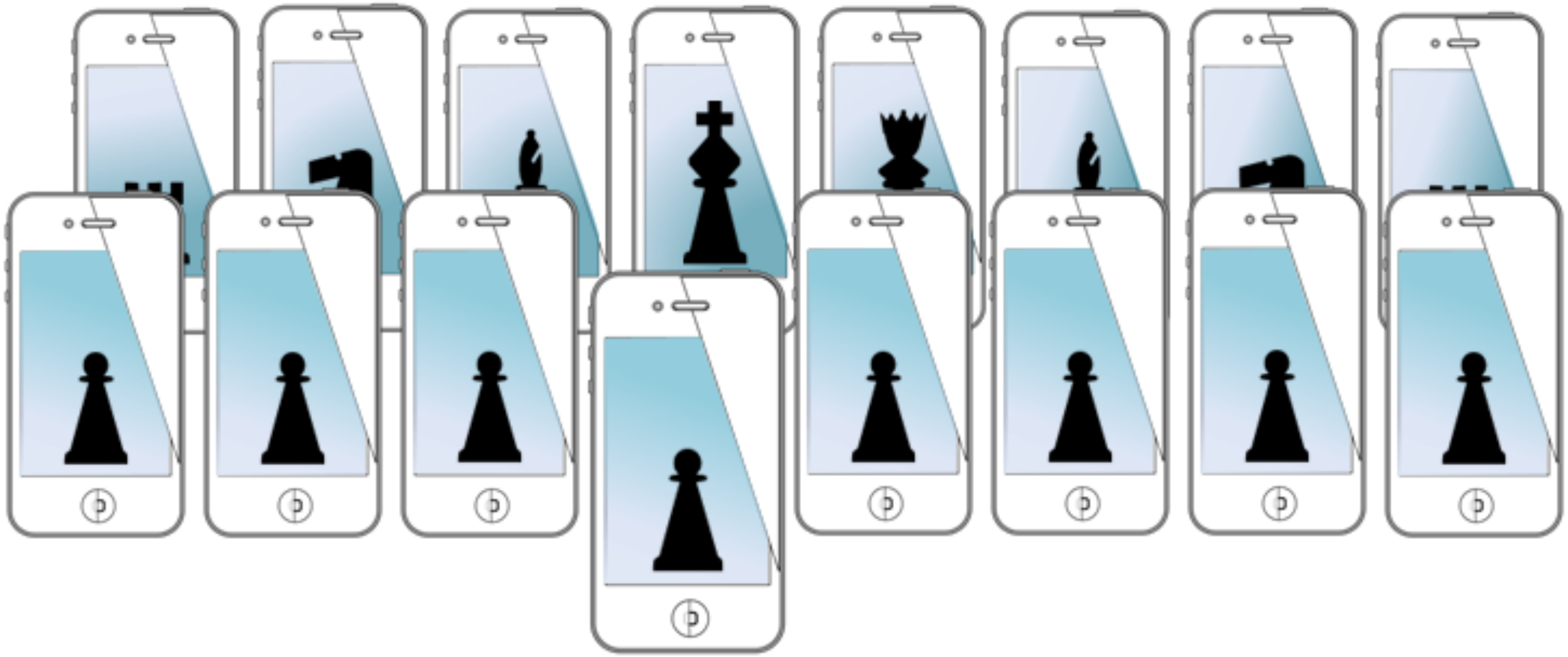
MOBILE SENSING LEARNING



CS5323 & 7323

Mobile Sensing and Learning

Speech Recognition and Dictation

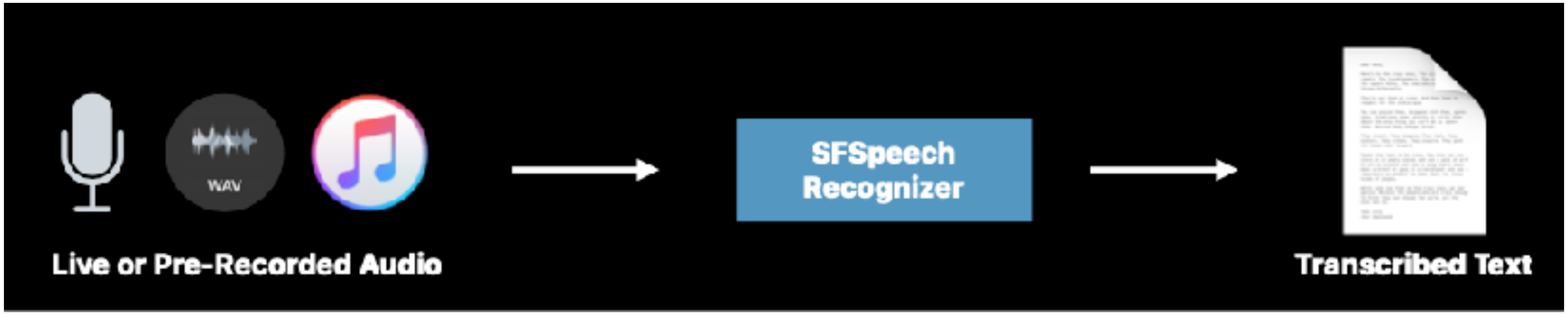
Eric C. Larson, Lyle School of Engineering,
Computer Science, Southern Methodist University

course logistics and agenda

- logistics:
  - final projects coming soon!!
- agenda:
  - speech recognition (from Speech Framework, SF)
  - SFSpeechRecognizer
    - **I helped Apple wreck a nice beach!**

overview

- introduced in 2016, same technology underpinning Siri
  - user must provide explicit authorization
  - free, but limited to certain number of recognitions per day
  - only allowed to dictate about 1 minute of audio
  - supports streaming audio and file I/O

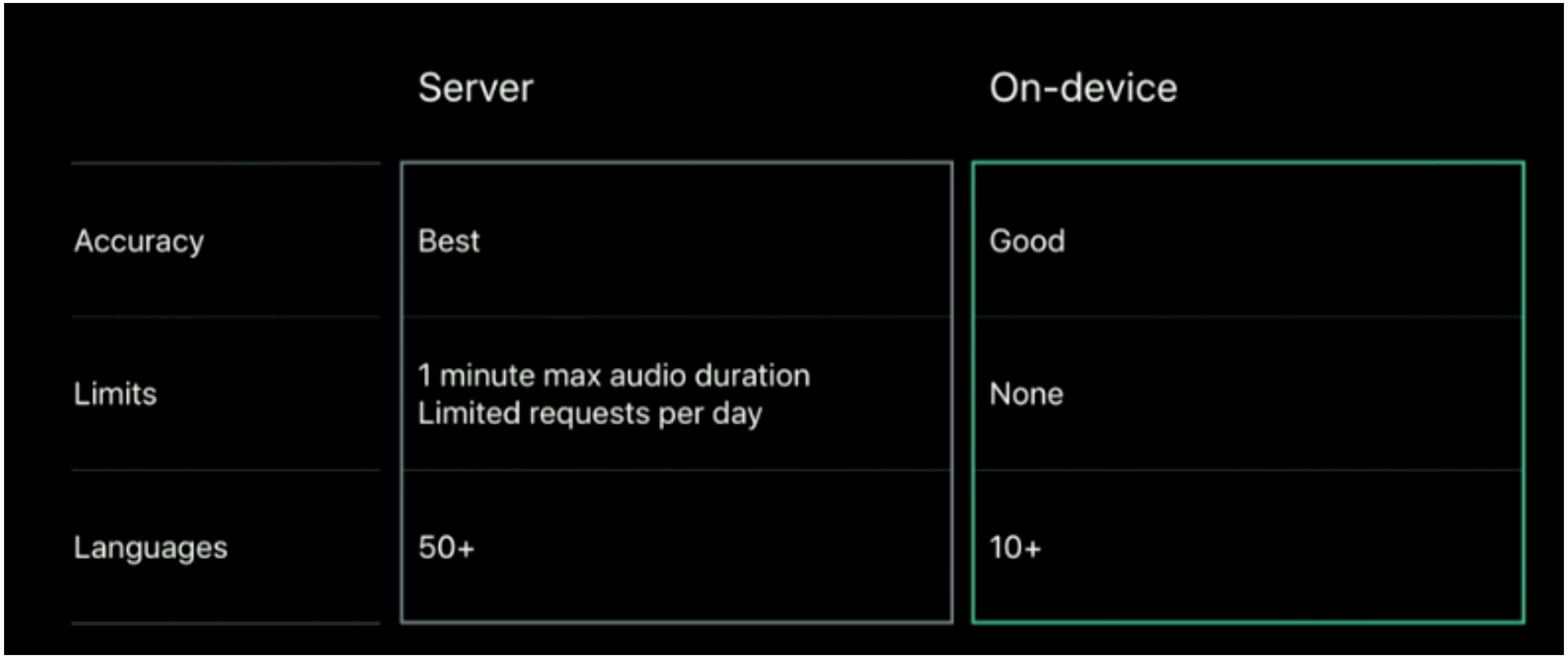


general usage

- uses API similar to REST (like Core Vision)
    - create a task
    - configure it (options)
    - start task
    - use completion handler (for updates and final text)
- best practices:
    - signify to user that the app is recording
    - show dictation as it happens

tradeoff server/on-device

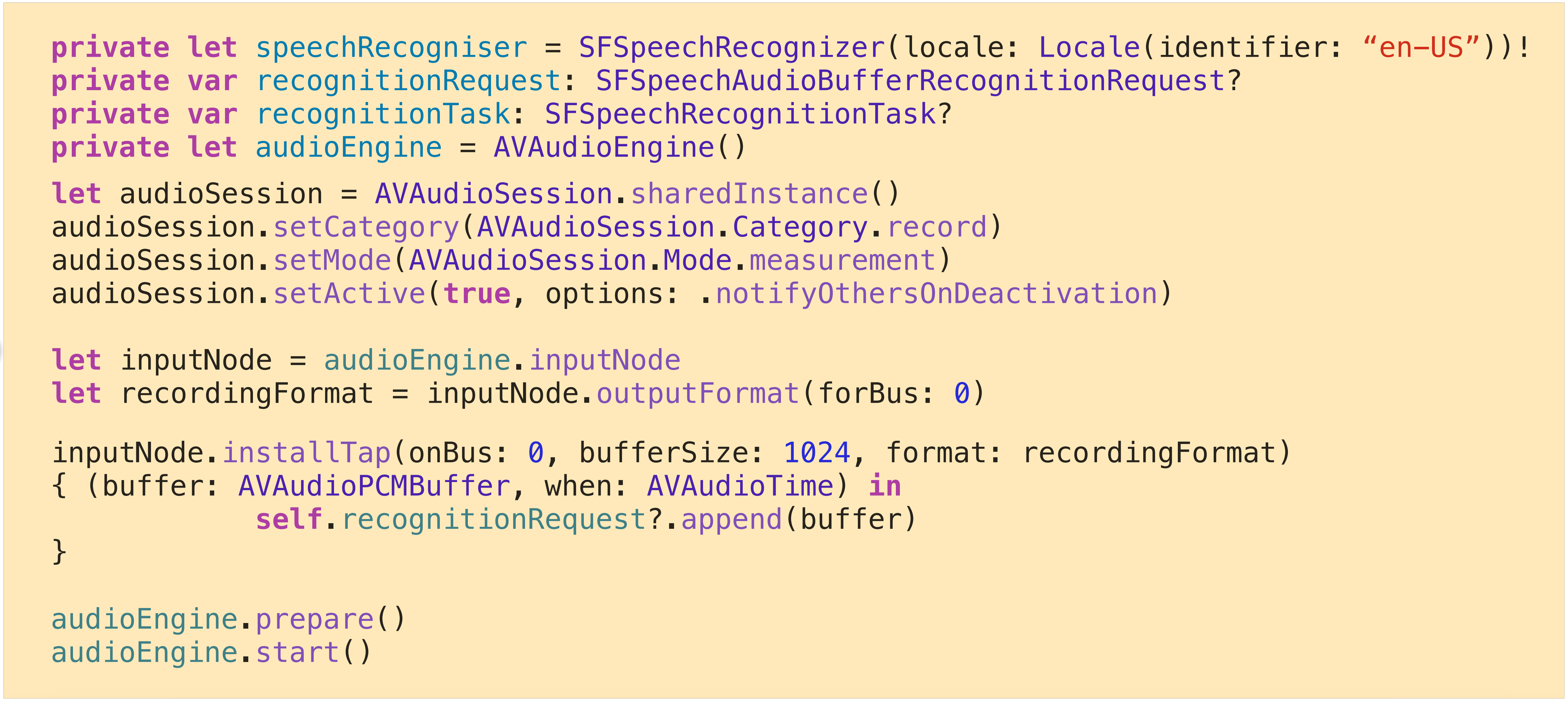
- 2016: speech is translated via cloud services
- 2019: also available on device



	Server	On-device
Accuracy	Best	Good
Limits	1 minute max audio duration Limited requests per day	None
Languages	50+	10+

using SFSpeechRecognizer

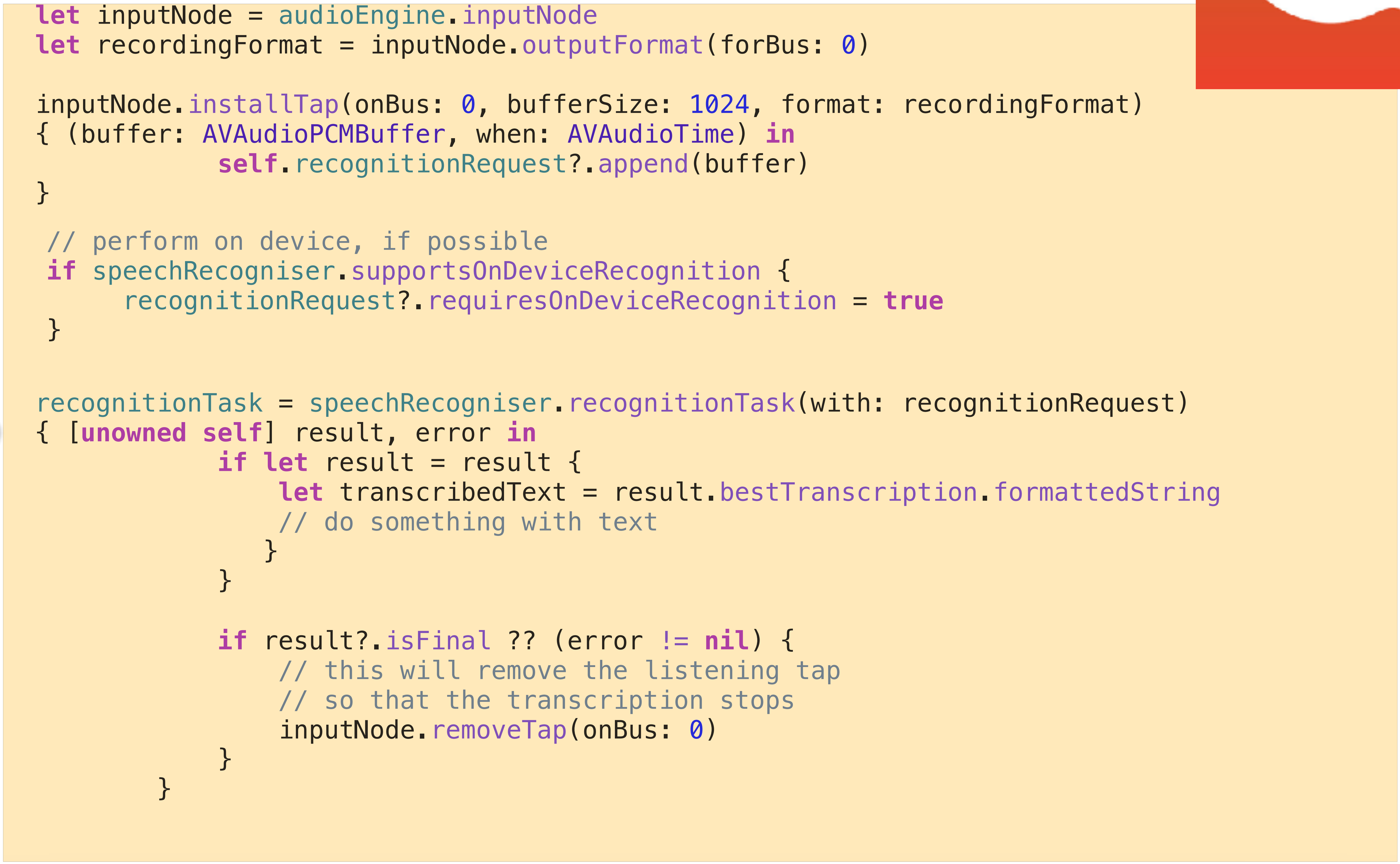
- callback model
- needs AVFoundation for adding audio chunks



```swift
private let speechRecogniser = SFSpeechRecognizer(locale: Locale(identifier: “en-US”))!
private var recognitionRequest: SFSpeechAudioBufferRecognitionRequest?
private var recognitionTask: SFSpeechRecognitionTask?
private let audioEngine = AVAudioEngine()

let audioSession = AVAudioSession.sharedInstance()
audioSession.setCategory(AVAudioSession.Category.record)
audioSession.setMode(AVAudioSession.Mode.measurement)
audioSession.setActive(true, options: .notifyOthersOnDeactivation)

let inputNode = audioEngine.inputNode
let recordingFormat = inputNode.outputFormat(forBus: 0)

inputNode.installTap(onBus: 0, bufferSize: 1024, format: recordingFormat)
{ (buffer: AVAudioPCMBuffer, when: AVAudioTime) in
            self.recognitionRequest?.append(buffer)
}

audioEngine.prepare()
audioEngine.start()
```

much of the code also has **guards** for **error checking**

using SFSpeechRecognizer



```
let inputNode = audioEngine.inputNode
let recordingFormat = inputNode.outputFormat(forBus: 0)

inputNode.installTap(onBus: 0, bufferSize: 1024, format: recordingFormat)
{ (buffer: AVAudioPCMBuffer, when: AVAudioTime) in
            self.recognitionRequest?.append(buffer)
}

 // perform on device, if possible
 if speechRecogniser.supportsOnDeviceRecognition {
     recognitionRequest?.requiresOnDeviceRecognition = true
 }


recognitionTask = speechRecogniser.recognitionTask(with: recognitionRequest)
{ [unowned self] result, error in
            if let result = result {
                let transcribedText = result.bestTranscription.formattedString
                // do something with text
               }
            }

            if result?.isFinal ?? (error != nil) {
                // this will remove the listening tap
                // so that the transcription stops
                inputNode.removeTap(onBus: 0)
            }
        }
```

more advanced speech processing

- recognition result contains many aspects of the voice, including:
  - Transcribed text (as we have seen)
  - Alternate transcriptions
  - Confidence in result
  - Timing
  - Speaking rate
  - Pause duration
  - Voice analytics

more advanced speech processing

```
bestTranscription=<SFTranscription>,
formattedString=I helped Apple recognize speech,

speakingRate=0.000000, averagePauseDuration=0.000000,

segments=(
    <SFTranscriptionSegment>, substringRange={0, 1}, timestamp=0.54, duration=0.24,
    confidence=0.966,
    substring=I, alternativeSubstrings=(\n),
    phoneSequence=AY,
    ipaPhoneSequence=\U02c8a\U0361\U026a, voiceAnalytics=(null),

    <SFTranscriptionSegment>, substringRange={2, 6}, timestamp=0.78,
    duration=0.3600000000000001, confidence=0.966,
    substring=helped, alternativeSubstrings=(\n),
    phoneSequence=h EH l p t,
    ipaPhoneSequence=h.\U02c8\U025b.l.p.t,
    voiceAnalytics=(null),

    …

    <SFTranscriptionSegment>, substringRange={25, 31}, timestamp=2.49,
    duration=0.5699999999999998, confidence=0.966,
    substring=speech, alternativeSubstrings=(\n),
    phoneSequence=s p EE ch,
    ipaPhoneSequence=s.p.\U02c8i.t\U0361\U0283, voiceAnalytics=(null)
),
```

running on device will limit the available features!

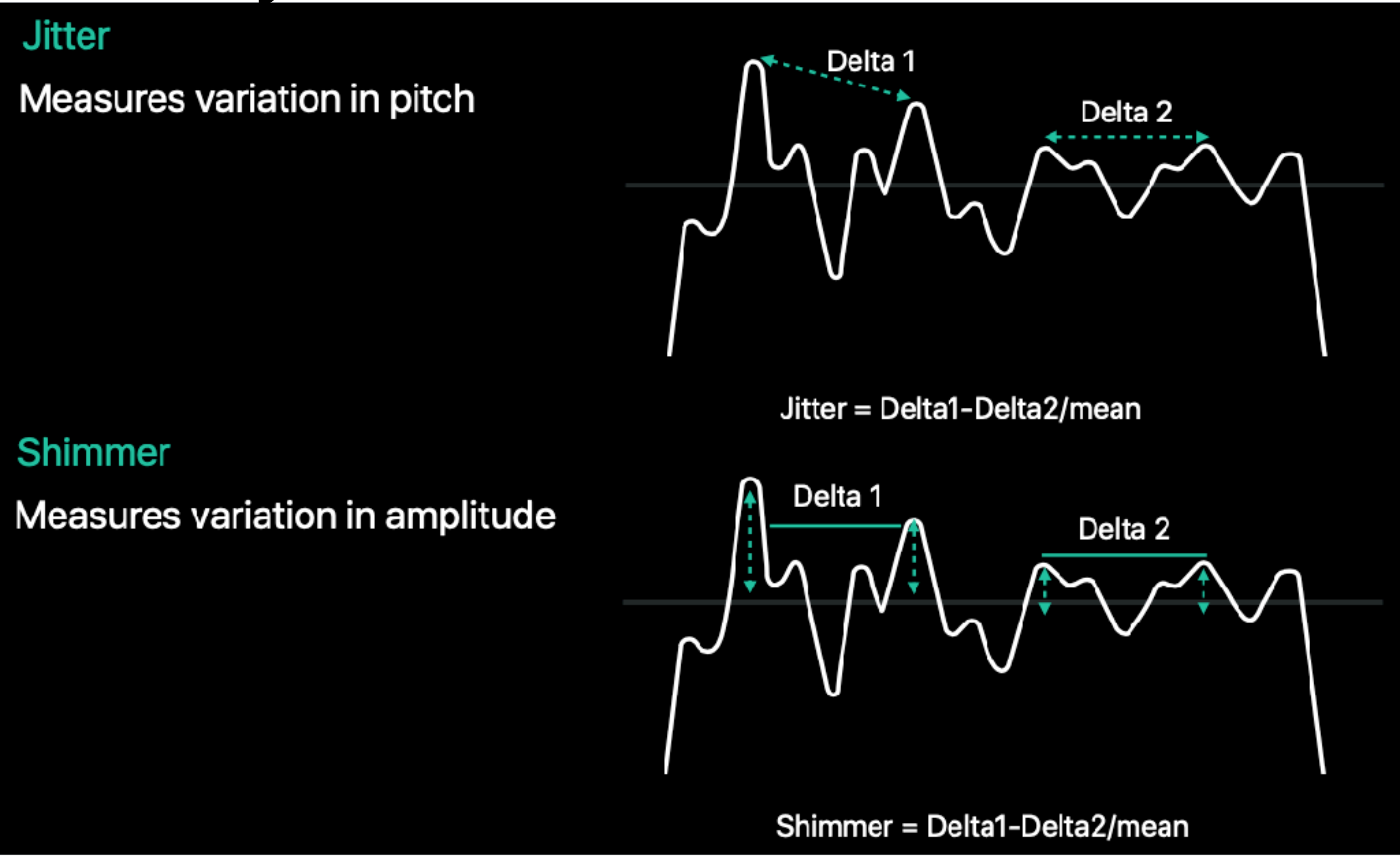
more advanced speech processing

Each segment now has incredible amount of information

```
<SFTranscriptionSegment>,
substringRange={0, 10}, timestamp=0.27, duration=0.65, confidence=0.911,
substring=Performing,
alternativeSubstrings=(\n),
phoneSequence=(null),
ipaPhoneSequence=(null),

voiceAnalytics=<SFVoiceAnalytics>,
jitter=<SFAcousticFeature>, featureValues=(12.53122 … 0.6218916),
frameDuration=0.010000,
shimmer=<SFAcousticFeature>, featureValues=(0.7158176 … 2.518468),
frameDuration=0.010000,
pitch=<SFAcousticFeature>, featureValues=(0.8526305,   …. 0.04258926),
frameDuration=0.010000,
voicing=<SFAcousticFeature>, featureValues=(0.07444749 …   0.4056852),
frameDuration=0.010000",
```

each featureValues array is length of
audio frames, could be **hundreds** of values

analytics



Jitter

Measures variation in pitch

Jitter = Delta1-Delta2/mean

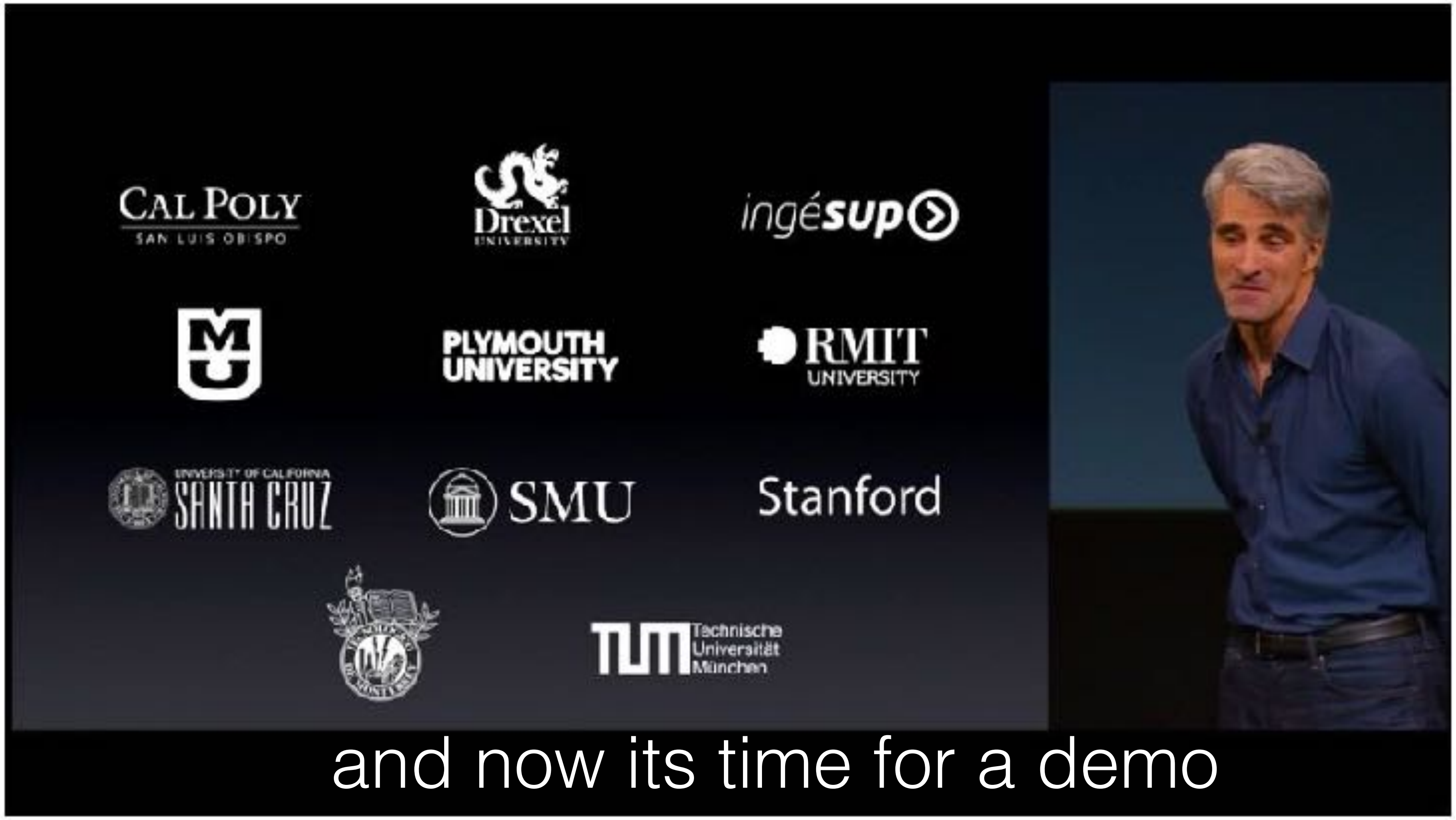
Shimmer

Measures variation in amplitude

Shimmer = Delta1-Delta2/mean

SFSpeechRecognizer

- adding audio blocks from input buffer



and now its time for a demo

Bonus: Create ML (iOS 15.0+)

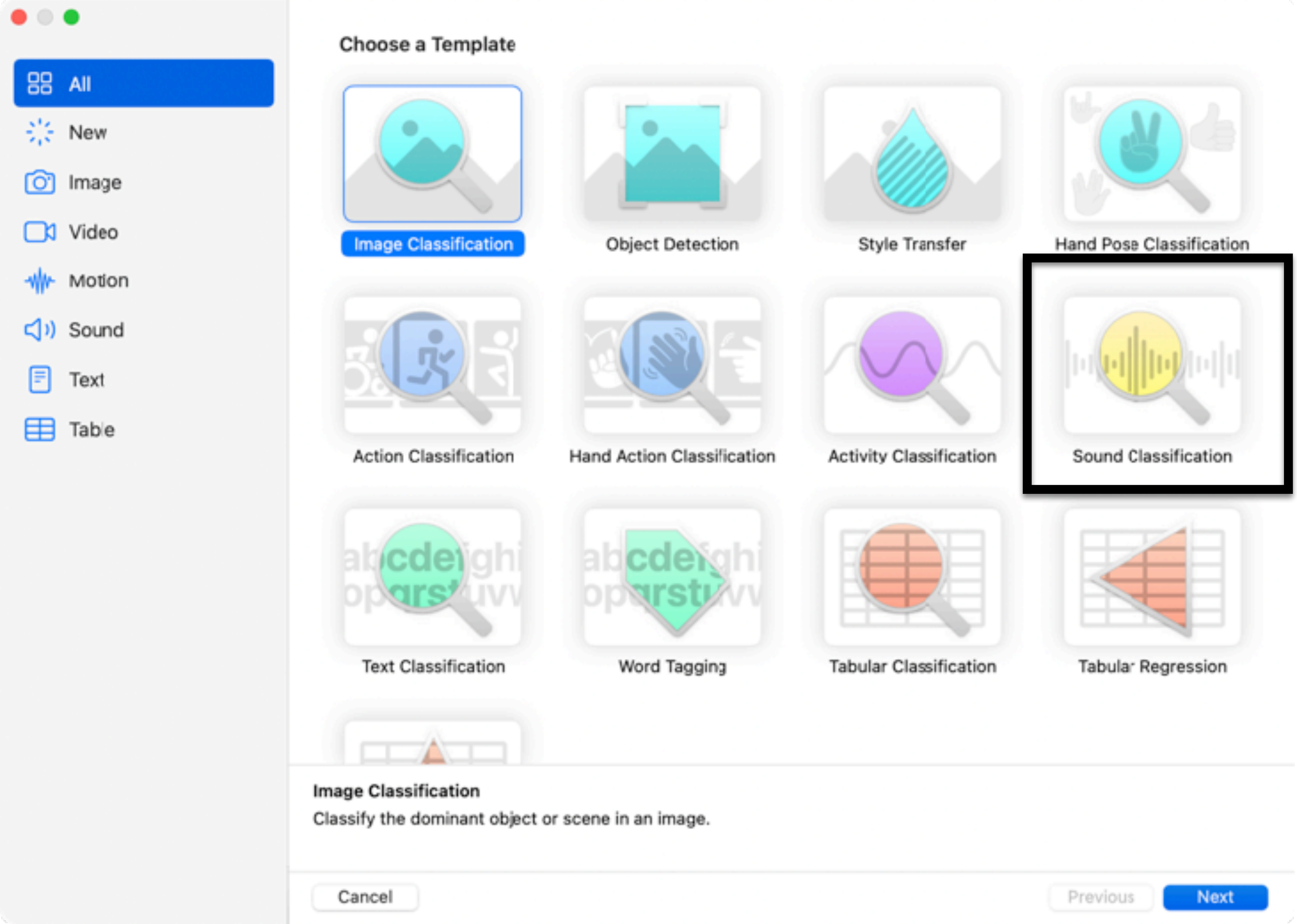
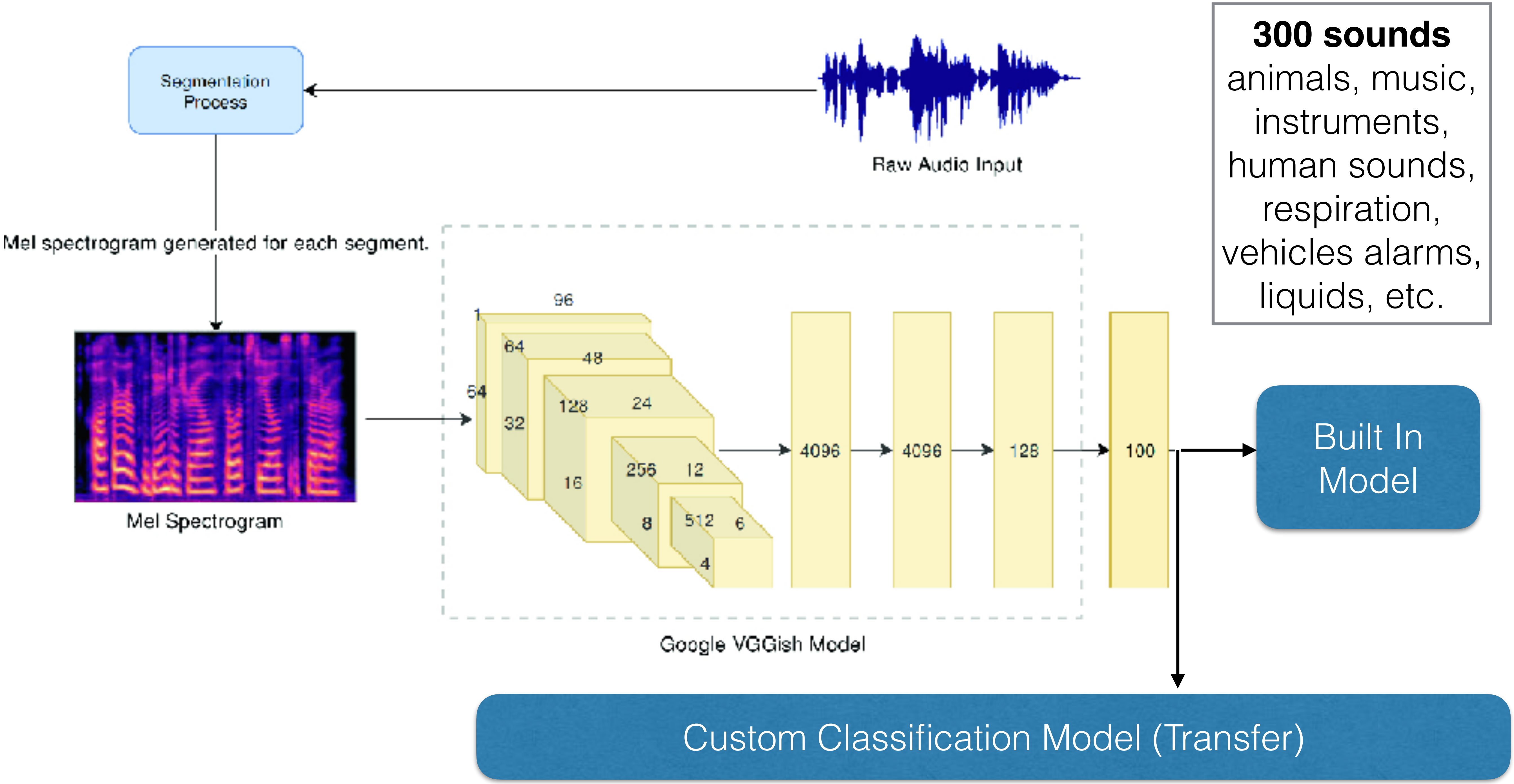


Image Classification

Classify the dominant object or scene in an image.

Create ML Audio Analyzer

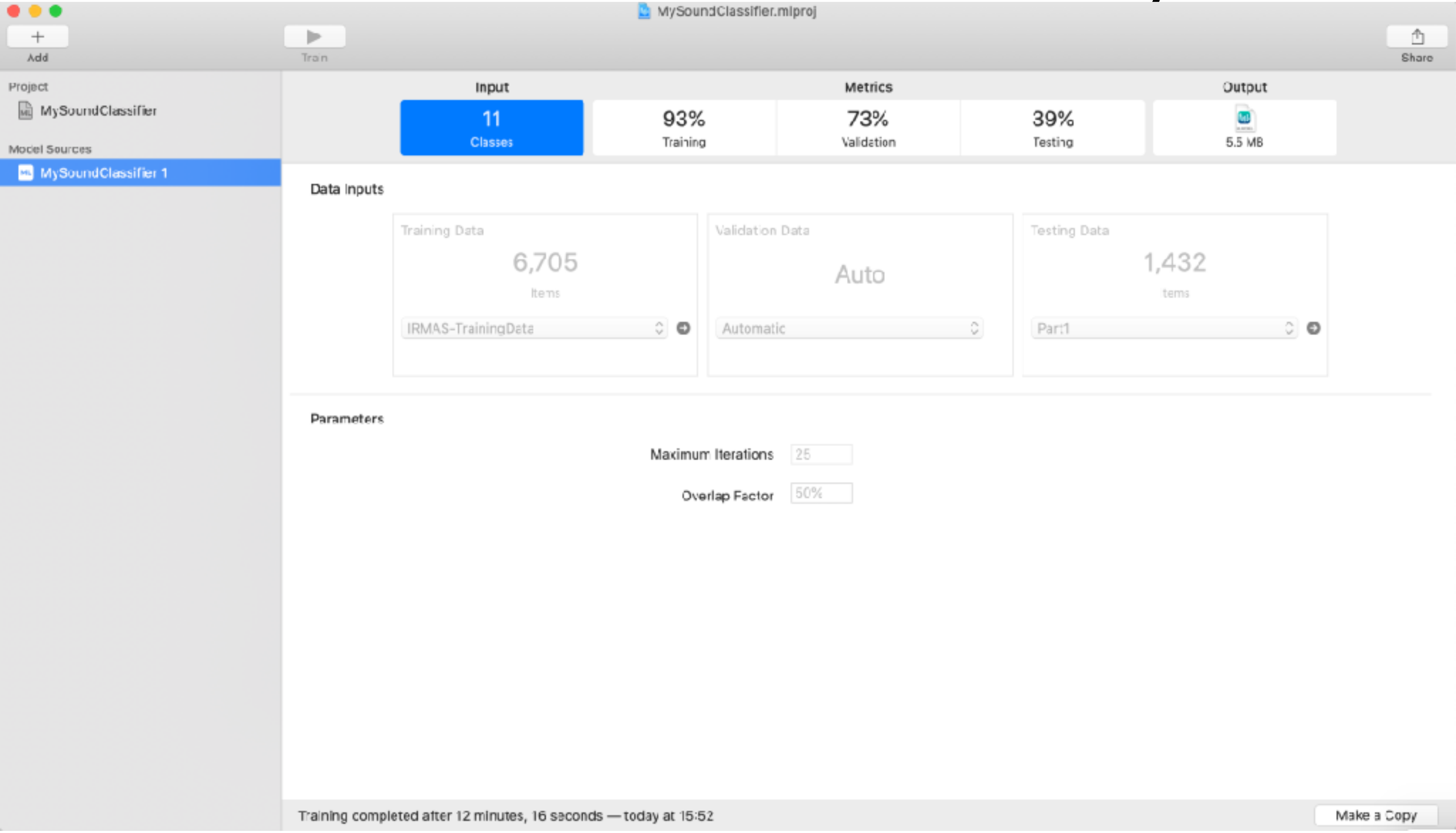


Built In Model

Custom Classification Model (Transfer)

300 sounds animals, music, instruments, human sounds, respiration, vehicles alarms, liquids, etc.

Create ML Audio Analyzer



11 Classes

93% Training

73% Validation

39% Testing

5.5 MB

Data Inputs

Training Data: 6,705 Items, IRMAS-TrainingData

Validation Data: Auto, Automatic

Testing Data: 1,432 Items, Part1

Parameters

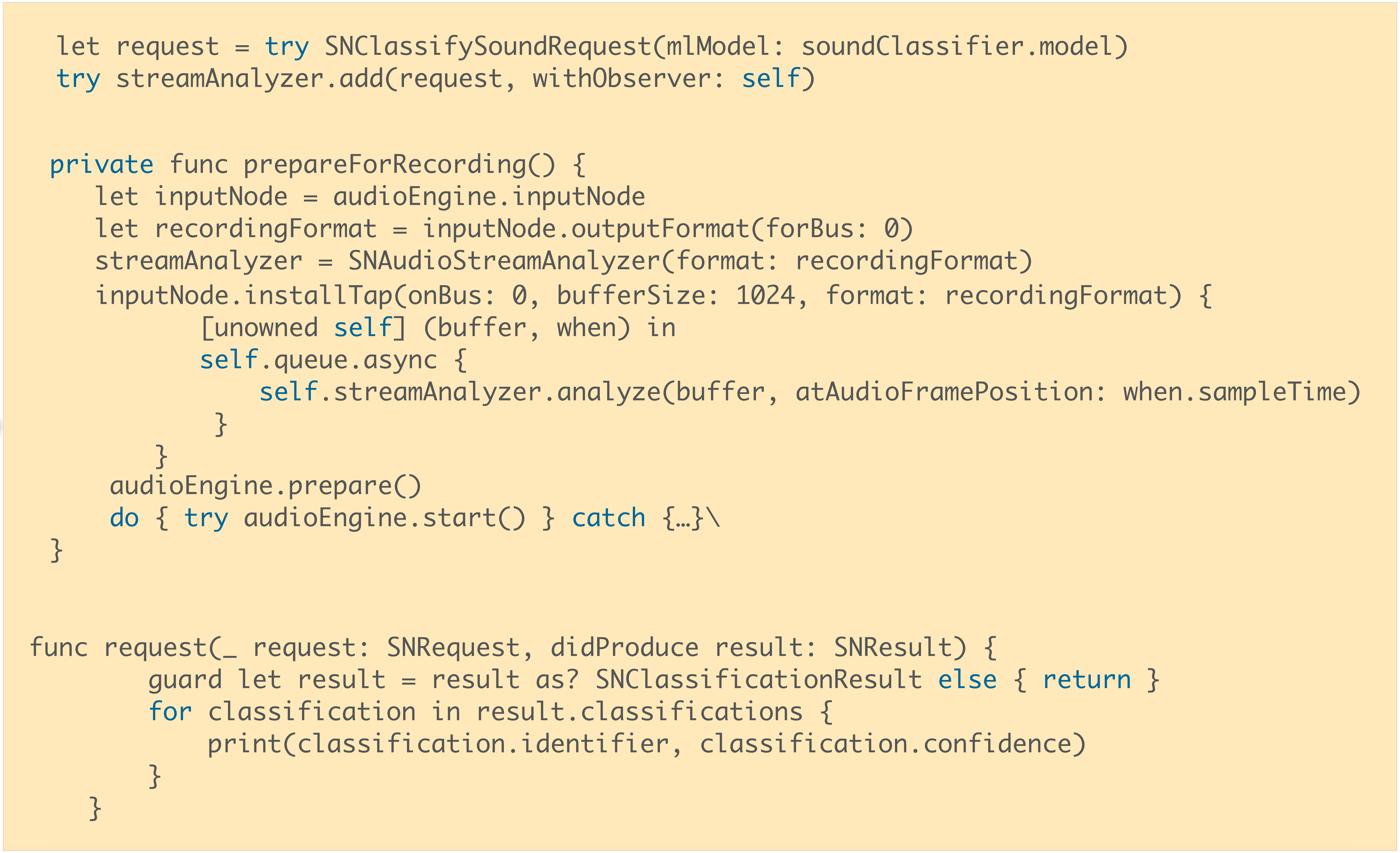
Maximum Iterations: 25

Overlap Factor: 50%

Training completed after 12 minutes, 16 seconds — today at 15:52

https://martinmitrevski.com/2019/12/09/sound-classification-with-create-ml-on-ios-13/

Create ML Audio Analyzer

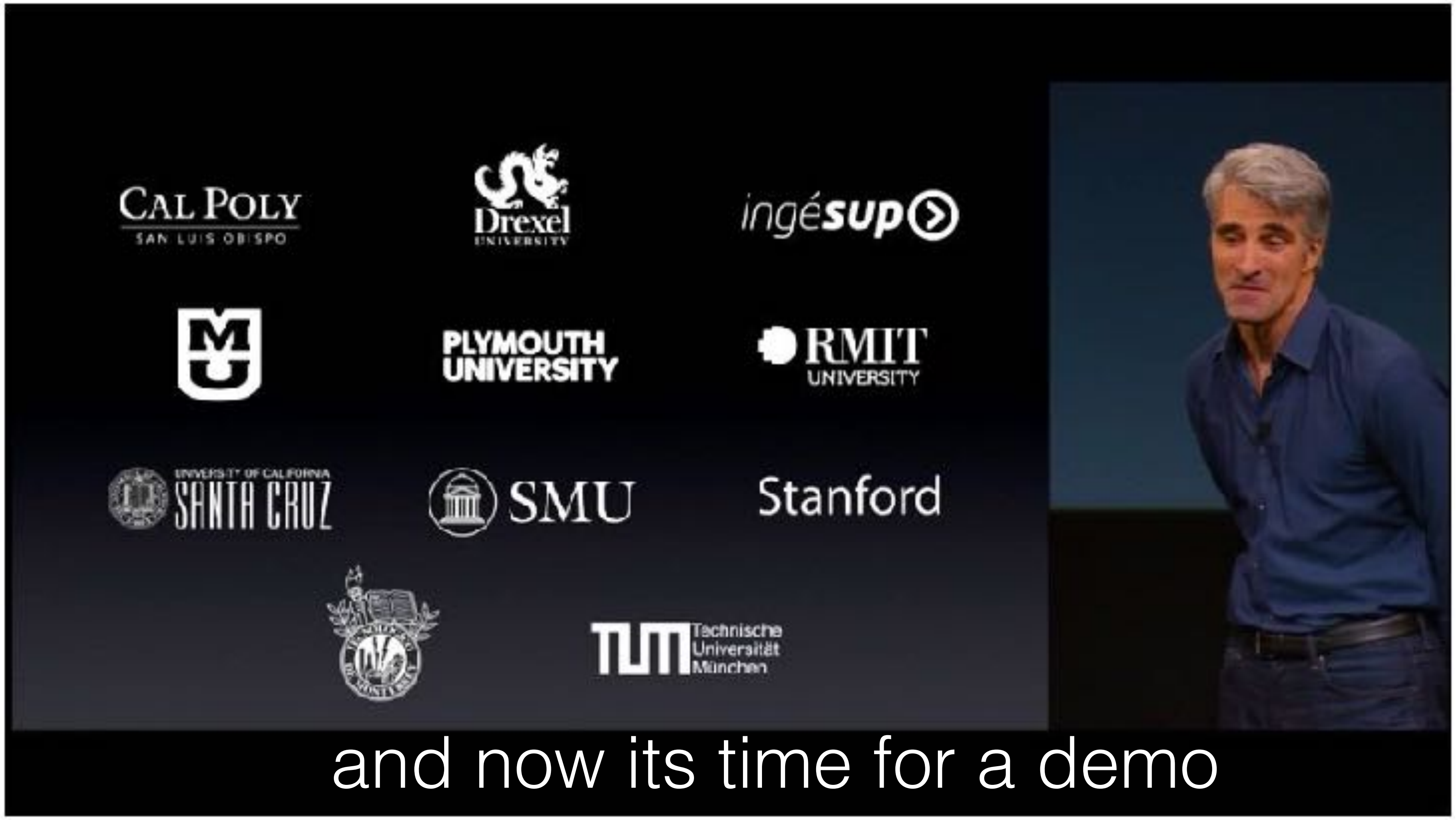


```
  let request = try SNClassifySoundRequest(mlModel: soundClassifier.model)
  try streamAnalyzer.add(request, withObserver: self)


 private func prepareForRecording() {
    let inputNode = audioEngine.inputNode
    let recordingFormat = inputNode.outputFormat(forBus: 0)
    streamAnalyzer = SNAudioStreamAnalyzer(format: recordingFormat)
    inputNode.installTap(onBus: 0, bufferSize: 1024, format: recordingFormat) {
           [unowned self] (buffer, when) in
           self.queue.async {
               self.streamAnalyzer.analyze(buffer, atAudioFramePosition: when.sampleTime)
            }
        }
     audioEngine.prepare()
     do { try audioEngine.start() } catch {…}\
 }


func request(_ request: SNRequest, didProduce result: SNResult) {
        guard let result = result as? SNClassificationResult else { return }
        for classification in result.classifications {
            print(classification.identifier, classification.confidence)
        }
    }
```

https://martinmitrevski.com/2019/12/09/sound-classification-with-create-ml-on-ios-13/

Sound Analysis

- add sound classification to our project

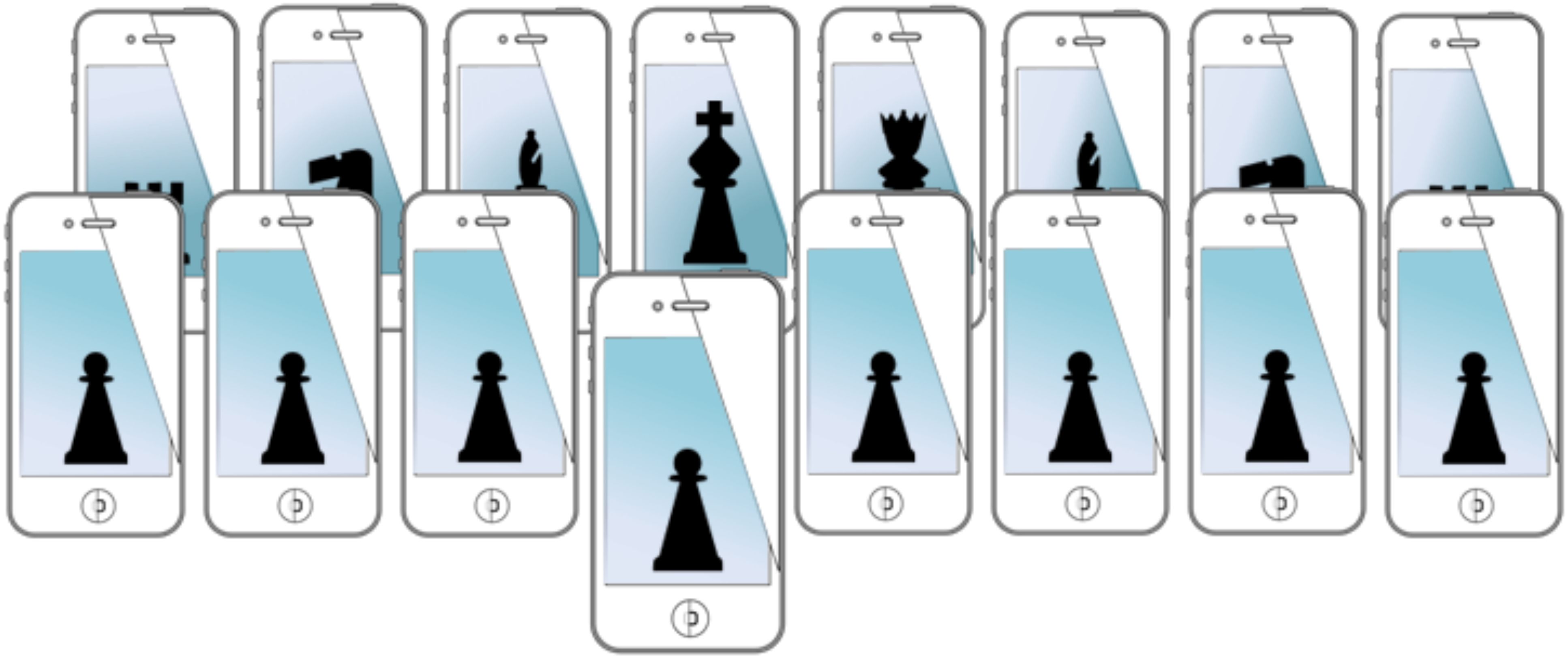


and now its time for a demo

for next time…

- Pitching
- ~Fin~

MOBILE SENSING LEARNING



CS5323 & 7323

Mobile Sensing and Learning

Speech Recognition and Dictation

Eric C. Larson, Lyle School of Engineering,
Computer Science, Southern Methodist University