1. Title Slide
2. Outline
3. Introduction
4. Motivation
   1. Smart devices are everywhere
   2. Many smart devices respond to voice commands
   3. Three big ones: Alexa, Siri, Google Assistant
   4. I use Google Assistant to control my smart home
   5. I am interested in embedded devices
   6. More importantly, I am interested in if they are always listening
5. Are they listening?
   1. Short answer: no
   2. Long answer: sort of..
6. Background
   1. Apple, Amazon, and Google have put a lot of research into wake words
   2. (Link to papers)
   3. Too computationally intensive to run voice recognition models all the time
   4. Train a smaller model to only pick up on wake words
   5. Once wake word is detected, following audio is processed by either onboard model or sent to the cloud for processing
   6. Many devices use dedicated processor just for wake words
   7. All of these devices use convolutional neural nets
   8. I’ll be using suggestions from
   9. Important step: MFTT calculations
      1. (Show what mel is)
7. Proposed Method
   1. Record audio samples of “Smithers”
   2. Download negative examples and noise from online datasets
   3. Synthesize data by overlaying positive and negative examples onto noise
   4. Trim data into 1 sec segments
   5. Perform MFTT calculations
   6. Train a CNN on the data
   7. Make predictions from sliding window
8. Preliminary Results
   1. Recorded 30 samples of “Smithers”
   2. Downloaded 125 negative examples from google dataset
      1. Words such as “dog” “seven” “forward” “go’
   3. Downloaded 200 5+ second background noises, such as rain, footsteps, wind
   4. Randomly selected one positive, one negative, and one background noise
   5. Selected random one second segment of background noise
   6. Synthesized positives: clean 1 second, noisy 1 second
   7. Synthesized negatives: clean 1 second, noisy 1 second
      1. Idea is to recognize and reject noise
         1. Idea similar to differential signals. No idea if legit
   8. Calculate MFTT on each
   9. Train CNN with positive and negative examples
   10. No sliding window so far
   11. Good validation and accuracy… or is it?
9. Discussion of prelimary results
   1. (Show examples of MFTTs)
   2. Model worked well recognizing when I said “Smithers”
      1. Keywords here “I said”, not smithers
      2. Recording new test cases for random words, results weren’t as good
      3. Perhaps model trained in on my voice and not of the word
   3. Need to play around with sampling, model parameters
10. Proposed research
    1. Play around is just what I will do
    2. Also need to develop sliding window for realtime audio
    3. Also need to explore implementation on low resource hardware
       1. How do I quantify limiting resources?
          1. Perhaps lower sample rates, model input paramters
11. Concluding remarks