

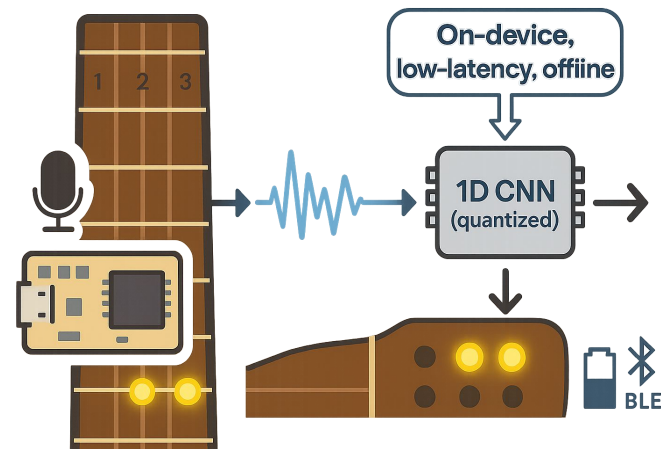


Guitar Helper

George Pan, Harvey Ko, Harrison Lo

Abstract

- Guitar Helper: an embedded ML system that listens to short audio snippets of strummed chords and classifies them in real time.



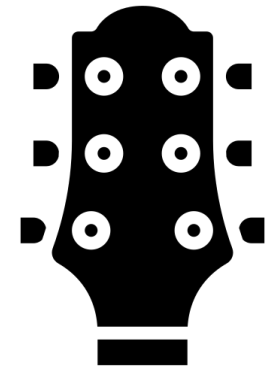
Objectives & Motivation



Beginners struggle with chord memorization and fast chord changes.

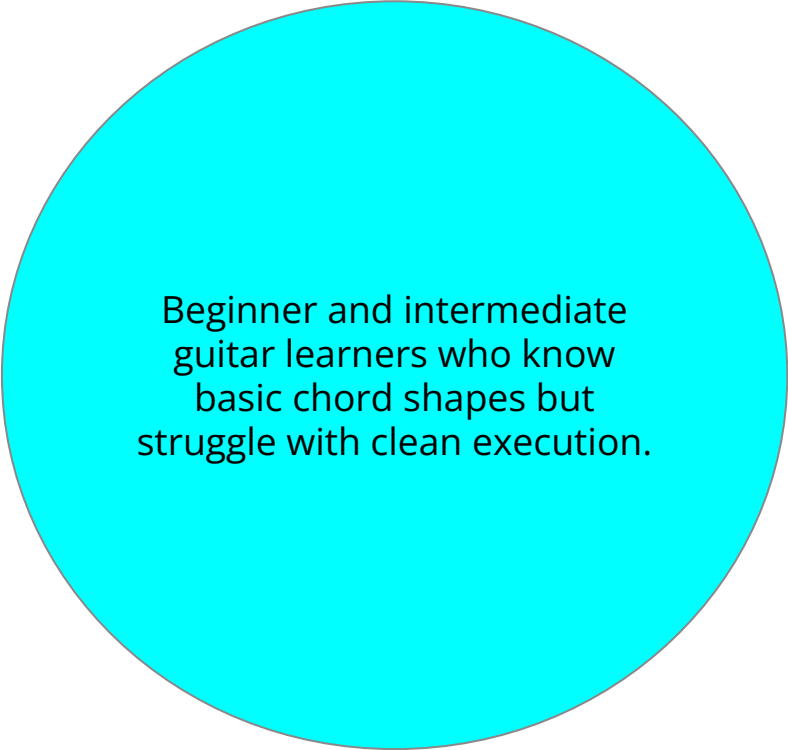


Difficult to find chord progressions online for less popular songs.

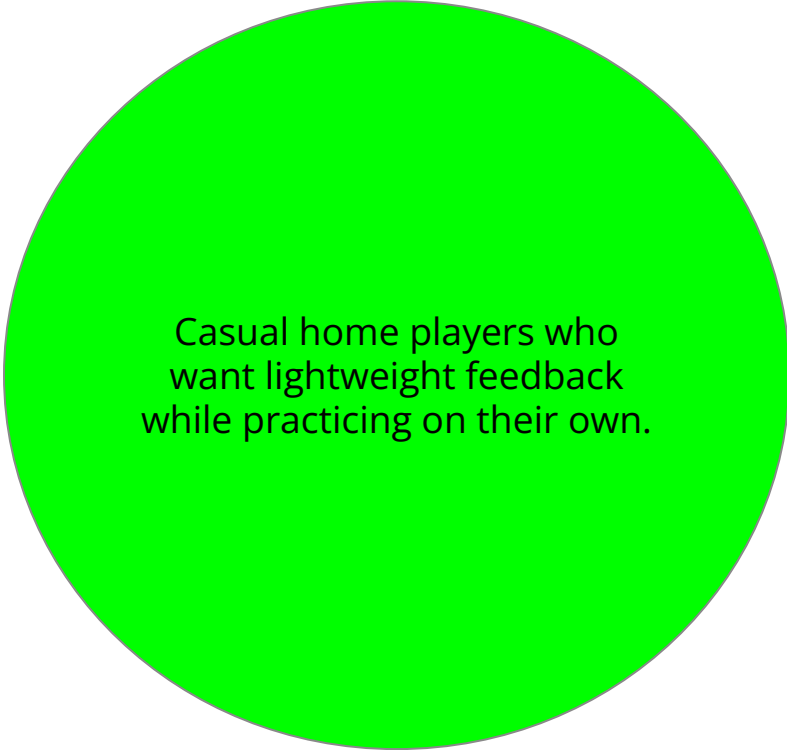


A palm-sized “Guitar Helper” that listens to songs and lights up the next chord on fretboard in real time using an embedded ML model.

Target Users



Beginner and intermediate guitar learners who know basic chord shapes but struggle with clean execution.



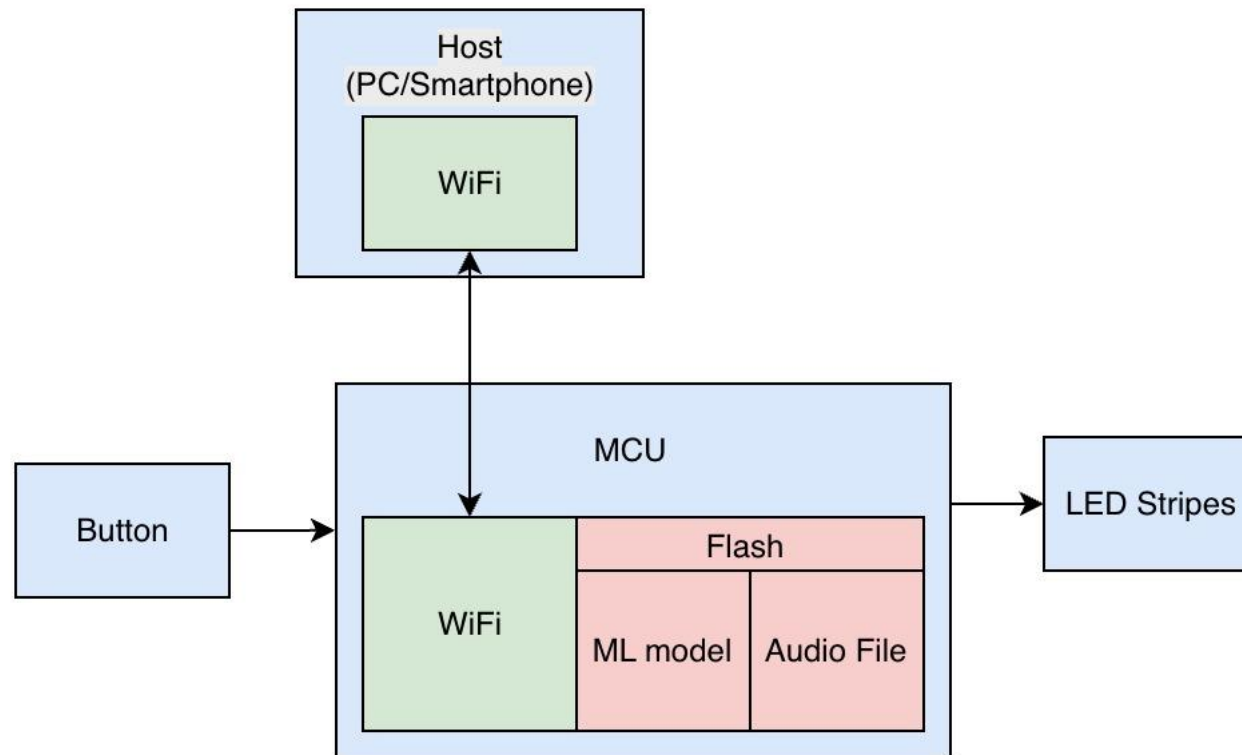
Casual home players who want lightweight feedback while practicing on their own.



BLERP Model

- Bandwidth
 - No internet connection is needed
- Latency
 - Recognize chords in real-time for better user experience
- Economics
 - No need for cloud or subscription services to generate chord from music file
- Reliability
 - No internet connection needed
- Privacy
 - Unpublished songs will not be exposed to internet

Overall block diagram





Data Collection & Pre-Processing

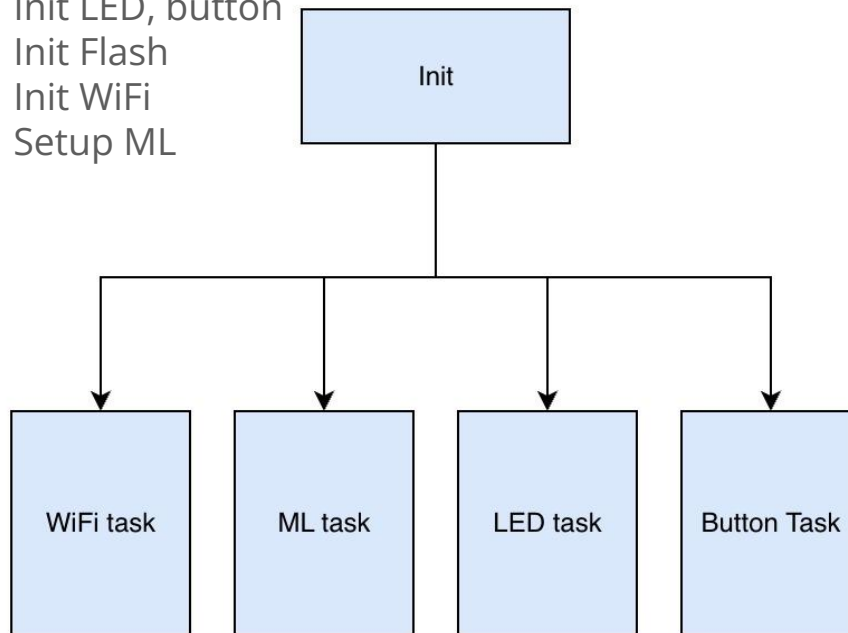
- Open source dataset from Kaggle ([Guitar Chords V3](#)).
- Preprocessing
 - Cropped each sample to a fixed length of 2 seconds.
 - The program will find the segment of 2 seconds with the largest volume.
- **We used RAW data as the input to our model.**

Machine Learning

Input layer (32,000 features)	Training settings	
Reshape layer (64 columns)	Number of training cycles ?	300
1D conv / pool layer (16 filters, 5 kernel size, 1 layer)	Use learned optimizer ?	<input type="checkbox"/>
Dropout (rate 0.25)	Learning rate ?	0.001
1D conv / pool layer (8 filters, 3 kernel size, 1 layer)	Training processor ?	GPU
Dropout (rate 0.25)	Advanced training settings	
Flatten layer	Validation set size ?	20 %
Add an extra layer	Split train/validation set on metadata key ?	
Output layer (8 classes)	Batch size ?	128
	Auto-weight classes ?	<input checked="" type="checkbox"/>
	Profile int8 model ?	<input checked="" type="checkbox"/>

Embedded System

1. Init LED, button
2. Init Flash
3. Init WiFi
4. Setup ML



Control Flow:

1. WiFi receives an audio file from the host and saves it to flash
2. ML task reads it out chunk by chunk (to reduce RAM usage) and performs chord recognition
3. Waits for user to press the start button, the fingering then shows up on the guitar, indicating by LEDs

Machine Learning Results

Last training performance (validation set)



ACCURACY
96.4%



LOSS
0.55

Confusion matrix (validation set)

	AM	BB	BDIM	C	DM	EM	F	G
AM	93.1%	0%	3.4%	0%	0%	3.4%	0%	0%
BB	0%	94.9%	5.1%	0%	0%	0%	0%	0%
BDIM	0%	2.7%	91.9%	0%	5.4%	0%	0%	0%
C	0%	0%	0%	100%	0%	0%	0%	0%
DM	0%	0%	3.0%	0%	97.0%	0%	0%	0%
EM	0%	0%	0%	0%	0%	100%	0%	0%
F	0%	0%	0%	3.6%	0%	0%	96.4%	0%
G	0%	0%	0%	2.3%	0%	0%	0%	97.7%
F1 SCORE	0.96	0.96	0.91	0.98	0.96	0.98	0.98	0.99

Validation Results



ACCURACY
94.71%

Metrics for Classifier



METRIC	VALUE
Area under ROC Curve ⓘ	0.99
Weighted average Precision ⓘ	0.96
Weighted average Recall ⓘ	0.96
Weighted average F1 score ⓘ	0.96

Confusion matrix

	AM	BB	BDIM	C	DM	EM	F	G	UNCERTAIN
AM	91.3%	0%	0%	2.2%	0%	4.3%	0%	0%	2.2%
BB	2.6%	92.3%	2.6%	0%	0%	0%	0%	0%	2.6%
BDIM	0%	2.1%	91.7%	0%	2.1%	2.1%	0%	0%	2.1%
C	0%	0%	0%	100%	0%	0%	0%	0%	0%
DM	2.4%	0%	0%	0%	92.9%	0%	0%	2.4%	2.4%
EM	2.1%	0%	0%	0%	0%	97.9%	0%	0%	0%
F	2.2%	0%	0%	2.2%	0%	0%	95.7%	0%	0%
G	0%	0%	2.3%	0%	0%	0%	0%	95.3%	2.3%
F1 SCORE	0.91	0.95	0.94	0.98	0.95	0.96	0.98	0.96	

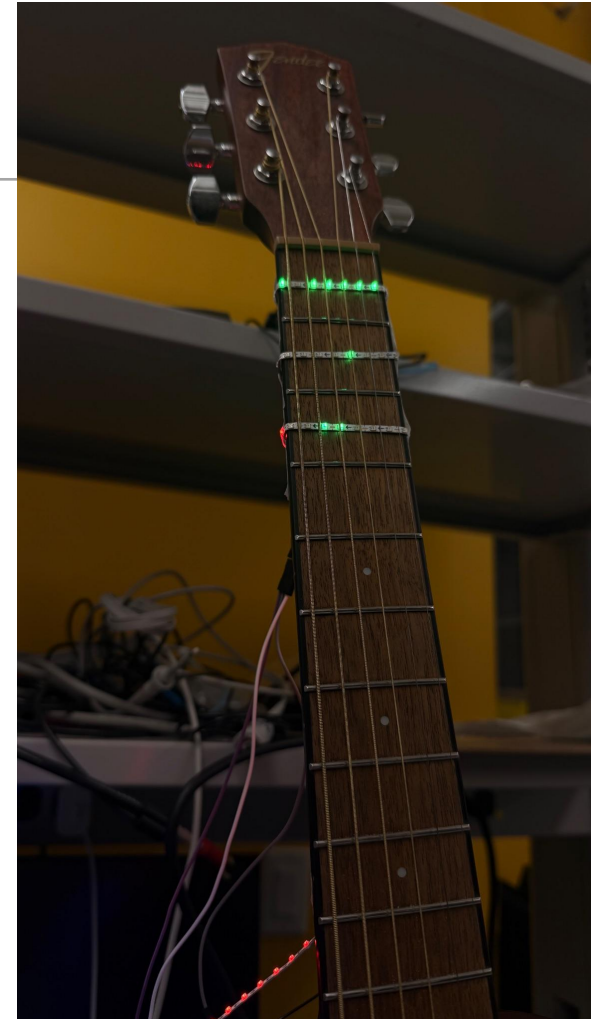
Test Results

Deployment

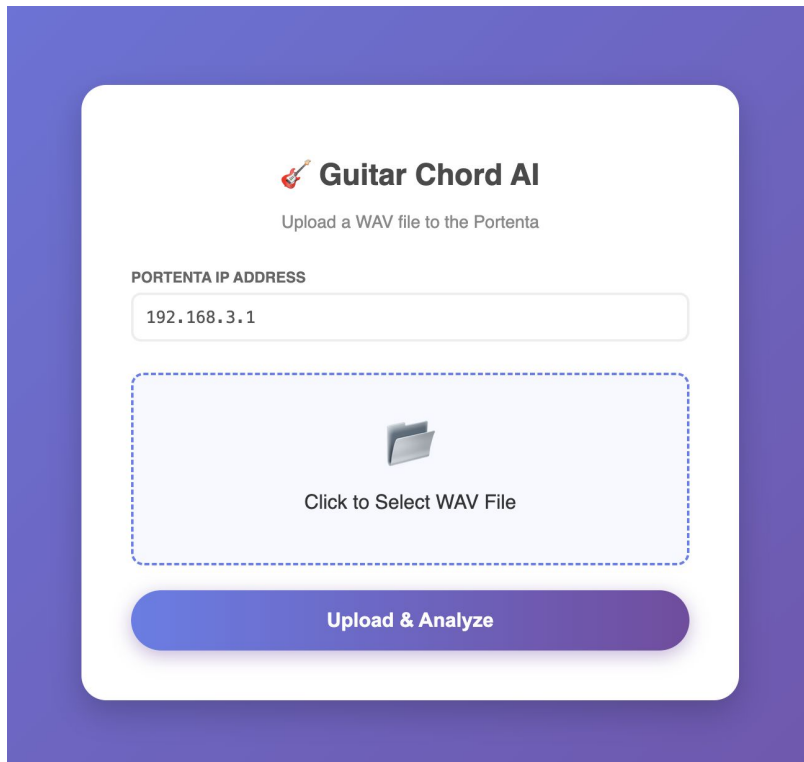
- Process 2s slice with 0.25 second hop
- Majority voting with window of 4, stride of 2
- Display LED chord patterns on guitar
 - Color code different finger (index, middle, ring, pinky)

```
20:45:52.873 -> --- Slice 1/101 (0.0000s - 2.0000s) ---
20:45:52.972 -> Am: 0.000000
20:45:52.972 -> Bb: 0.000000
20:45:53.006 -> Bdim: 0.000000
20:45:53.006 -> C: 0.996094
20:45:53.006 -> Dm: 0.000000
20:45:53.006 -> Em: 0.000000
20:45:53.006 -> F: 0.000000
20:45:53.006 -> G: 0.000000
20:45:53.006 -> >>> Detected: C (99.6%)
20:45:53.006 ->
20:45:53.006 -> --- Slice 2/101 (0.2500s - 2.2500s) ---
20:45:53.104 -> Am: 0.000000
20:45:53.104 -> Bb: 0.000000
20:45:53.104 -> Bdim: 0.000000
20:45:53.104 -> C: 0.996094
20:45:53.141 -> Dm: 0.000000
20:45:53.141 -> Em: 0.000000
20:45:53.141 -> F: 0.000000
20:45:53.141 -> G: 0.000000
20:45:53.141 -> >>> Detected: C (99.6%)
20:45:53.141 ->
```

```
20:46:08.204 -> 11.00s-13.75s: F (avg 81.2%)
20:46:08.204 -> 11.50s-14.25s: F (avg 79.6%)
20:46:08.204 -> 12.00s-14.75s: F (avg 80.5%)
20:46:08.204 -> 12.50s-15.25s: C (avg 97.8%)
20:46:08.204 -> 13.00s-15.75s: C (avg 99.3%)
20:46:08.204 -> 13.50s-16.25s: C (avg 99.6%)
20:46:08.204 -> 14.00s-16.75s: C (avg 99.6%)
20:46:08.204 -> 14.50s-17.25s: C (avg 99.6%)
20:46:08.204 -> 15.00s-17.75s: C (avg 99.6%)
20:46:08.204 -> 15.50s-18.25s: C (avg 99.6%)
20:46:08.204 -> 16.00s-18.75s: C (avg 99.5%)
20:46:08.204 -> 16.50s-19.25s: C (avg 99.5%)
20:46:08.204 -> 17.00s-19.75s: G (avg 88.8%)
20:46:08.204 -> 17.50s-20.25s: G (avg 86.7%)
20:46:08.204 -> 18.00s-20.75s: G (avg 93.9%)
```

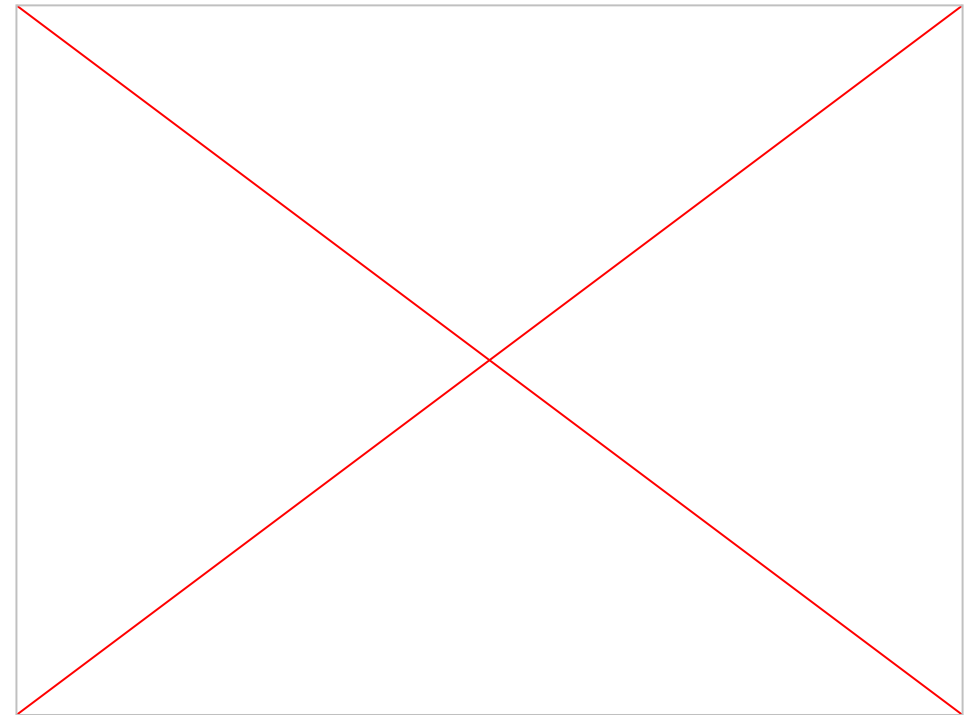


Deployment



The screenshot shows a web interface for 'Guitar Chord AI'. At the top, there is a guitar icon and the title 'Guitar Chord AI'. Below the title, it says 'Upload a WAV file to the Portenta'. There is a text input field labeled 'PORTENTA IP ADDRESS' containing the value '192.168.3.1'. Below this is a dashed blue box containing a folder icon and the text 'Click to Select WAV File'. At the bottom of the interface is a large blue button labeled 'Upload & Analyze'.

Audio Upload Page



Chord Pattern LED on Guitar Fretboard



Challenges

- Software
 - Selective to audio file
 - works well with clear guitar audio
 - struggles with songs that have several instruments or effects
- Hardware
 - Making sure LED strips do not interfere with guitar strings
 - Limited storage for audio files



Thank you