Objective

Attempting to search for sheet music t favorite song is difficult enough, but buy music can also become quite expensive c With that in mind, we would like to give learners, piano players specifically, a bette of learning songs. The goal of this proje create a system that takes video or audio a piano tutorial and outputs sheet m





MIDI i essent music compl Basic create Spotify machi learnir Pytho to dea audio

Sheet Music Generator

Group members: Alan Zhao '23, Isaiah Curry '23

Faculty advisors: Dr. Kevin Huang, Ph.D., Dr. Lin Cheng, Ph.D.

	Methodolo	
nusic to one's out buying sheet nsive over time. to give music a better method is project is to audio input from heet music.	 user input of Mp4 to mp3 format) Audio to MI MIDI to shee We utilize a sing conversion, four conversion, and Musescore, a to 	f mp4/mp3 file conversion (if i DI conversion et music convers gular python scr scripts for audi a manual input ol for building s
		Code
MIDI is essentially sheet music for computers Basic Pitch, created by Spotify, utilizes machine learning and	<pre>In [4]: import os import sys from moviepy.editor import * mpffile = "/Users/isaiahcurry/downloads/capstonetestvid mp3file = "/Users/isaiahcurry/downloads/capstonetestvid vidclip = VideoFileClip(mpffile) audclip, elose vidclip.close NoviePy - Writing audio in /Users/isaiahcurry/downloads MoviePy - Done. Out[4]: Vound method VideoFileClip.close of <moviepy.video.io.< pre=""></moviepy.video.io.<></pre>	<pre>1</pre>
Python libraries	<pre>contours: np.ndarray, note_events: List[Tuple[int, int, int, float]], n_bins) -> List[Tuple[int, int, int, float, Optional[List[int]]]]: """Given note events and contours, estimate pitch bends per note. Pitch bends are represented as a sequence of evenly spaced midi pitch bend c The time stamps of each pitch bend can be inferred by computing an evenly sp the start and end times of each note event. Args: contours: Matrix of estimated pitch contours note_events: note event tuple Defended to the pitch bend to tuple Defended totuple Defended to tup</pre>	<pre>for audio_path in audio_path_list: print("") try: model_output, midi_data, note_events = predict(pathlib.Path(audio_path), model, onset_threshold, frame threshold.</pre>

logy

f input is in video

ersion

script for mp4 to mp3 udio to MIDI out of the MIDI file into g sheet music.

1. Mp4 to mp3 conversion (if necessary) 2. Creates note events

3. Creates MIDI arguments

> Computes 4. rhythm Makes 5.

> > off model, saves MIDI file





(**9:#2:**** **9 * * * ***

From our results, we can conclude that our MIDI data transcribed from the input audio is accurate, when solely looking into note accuracy. This note accuracy within our MIDI files translates to note accuracy within our sheet music. However, we run into issues when looking into other parameters such as time signature and tempo, where we have seen that we get inaccuracy.

Interestingly enough, with our program, we are not prediction based limited to piano. Our program should work with any instrument in theory, which includes even voice.

Results

- 1. Visual representation of output MIDI file (using Logic Pro) from sample piano audio
- Sheet music output 2. from sample piano audio

This sample is a six second electric piano audio clip

Conclusions