Final Project:
COZMO SINGS

Bonnie Guo & Fiona Chiu

Presentation Agenda

01

OVERVIEW

The problem that we are attempting to solve.

02

THE APPROACH

Our project idea and solution to the problem.

03

INSIGHTS

The most interesting aspects of our solution.

04

FUTURE PLANS

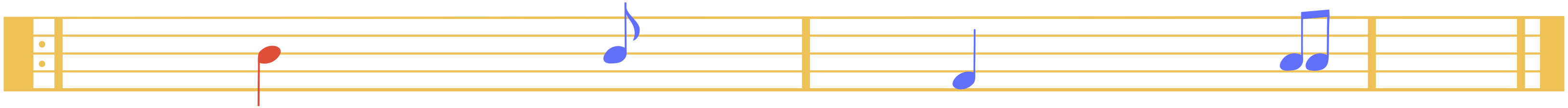
Potential extensions to our project.

Project

1. Use GPT to perform optical music recognition and have Cozmo sing songs.

1. Turn Cozmo into a piano – play notes on a keyboard that he sings out loud.

Our Pipeline



STEP 1

Use CV to threshold and augment the image captured by Cozmo.

STEP 2

Use CV to identify key elements (song title, staff lines, bar lines for measure segmentation)

STEP 3

Pass the segmented images into GPT-4’s vision model for note, title, and melody generation.

STEP 4

Convert GPT’s parsed string into Cozmo SongNote objects.

Our Pipeline

STEP 1

Use CV to threshold and augment the image captured by Cozmo.

STEP 2

Use CV to identify key elements (song title, staff lines, bar lines for measure segmentation)

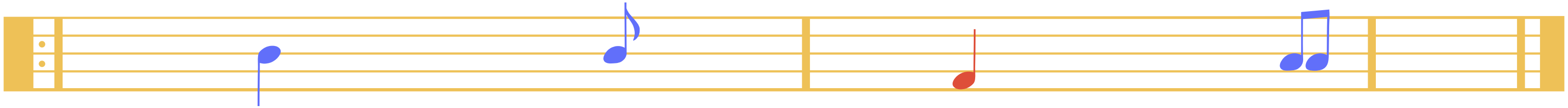
STEP 3

Pass the segmented images into GPT-4’s vision model for note, title, and melody generation.

STEP 4

Convert GPT’s parsed string into Cozmo SongNote objects.

Our Pipeline



STEP 1

Use CV to threshold and augment the image captured by Cozmo.

STEP 2

Use CV to identify key elements (song title, staff lines, bar lines for measure segmentation)

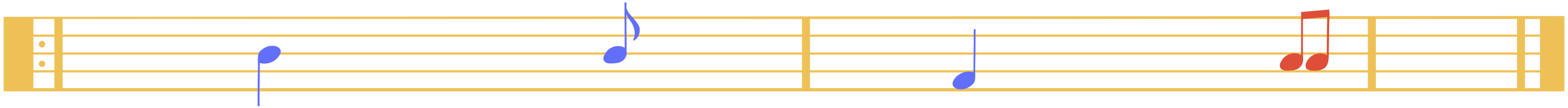
STEP 3

Pass the segmented images into GPT-4's vision model for note, title, and melody generation.

STEP 4

Convert GPT's parsed string into Cozmo SongNote objects.

Our Pipeline



STEP 1

Use CV to threshold and augment the image captured by Cozmo.

STEP 2

Use CV to identify key elements (song title, staff lines, bar lines for measure segmentation)

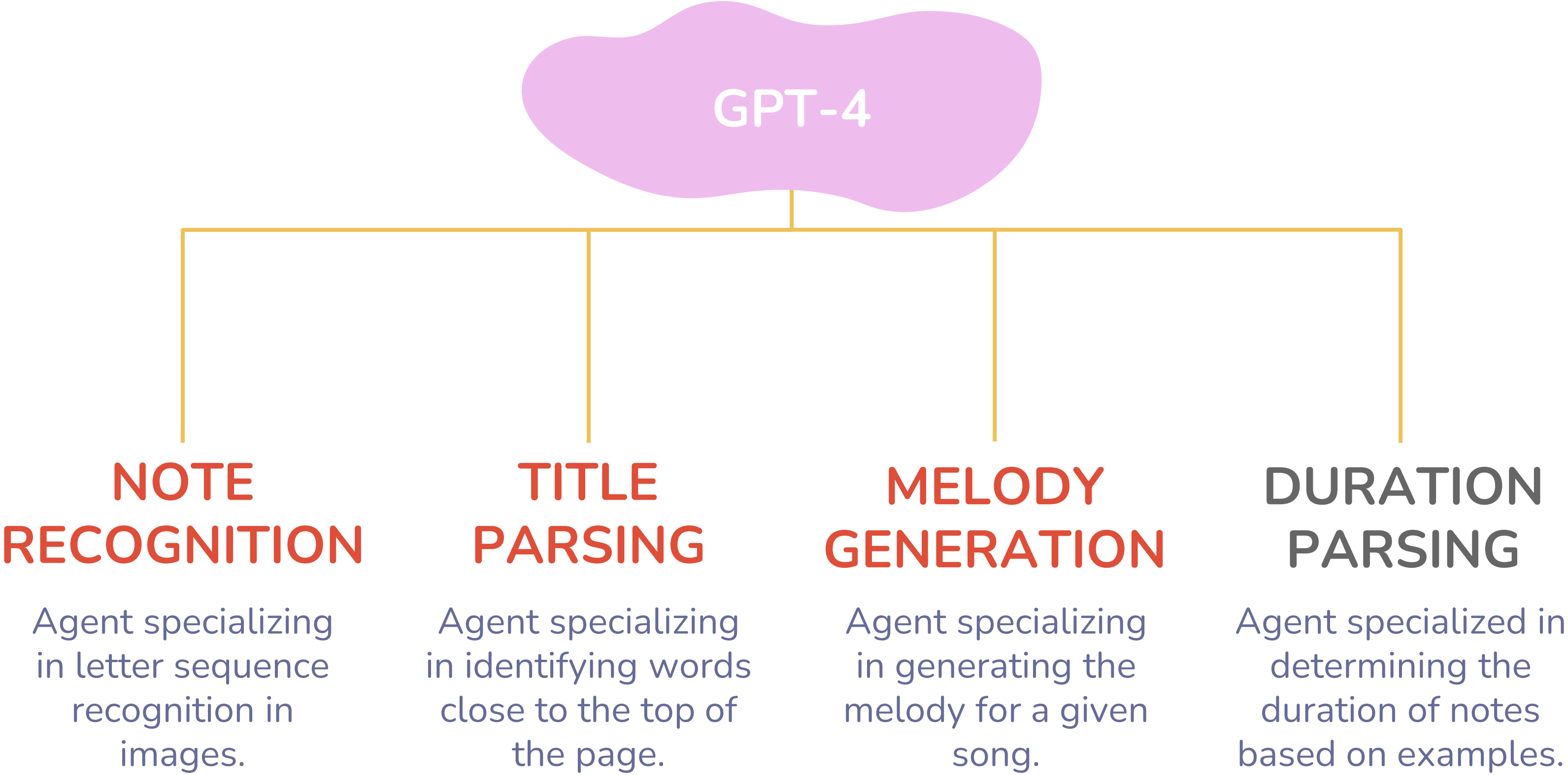
STEP 3

Pass the segmented images into GPT-4's vision model for note, title, and melody generation.

STEP 4

Convert GPT's parsed string into Cozmo SongNote objects.

Prompt Engineering



GPT-4

NOTE RECOGNITION

Agent specializing in letter sequence recognition in images.

TITLE PARSING

Agent specializing in identifying words close to the top of the page.

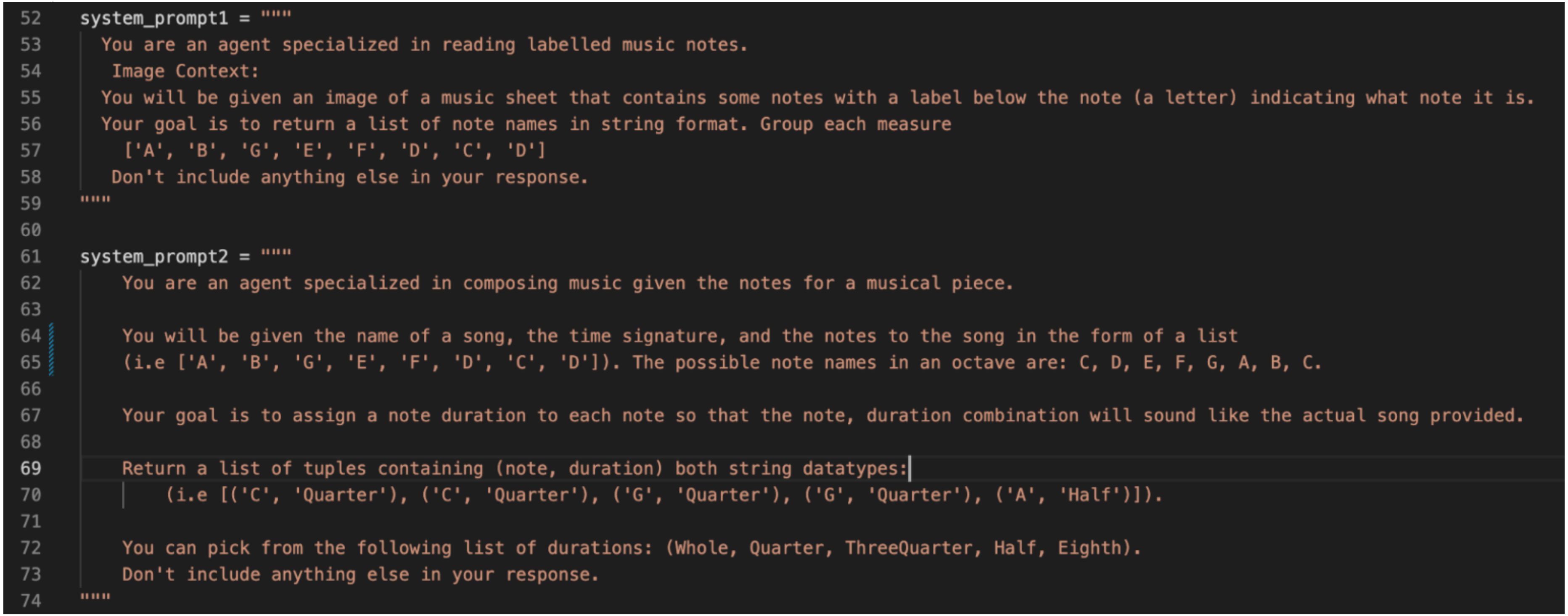
MELODY GENERATION

Agent specializing in generating the melody for a given song.

DURATION PARSING

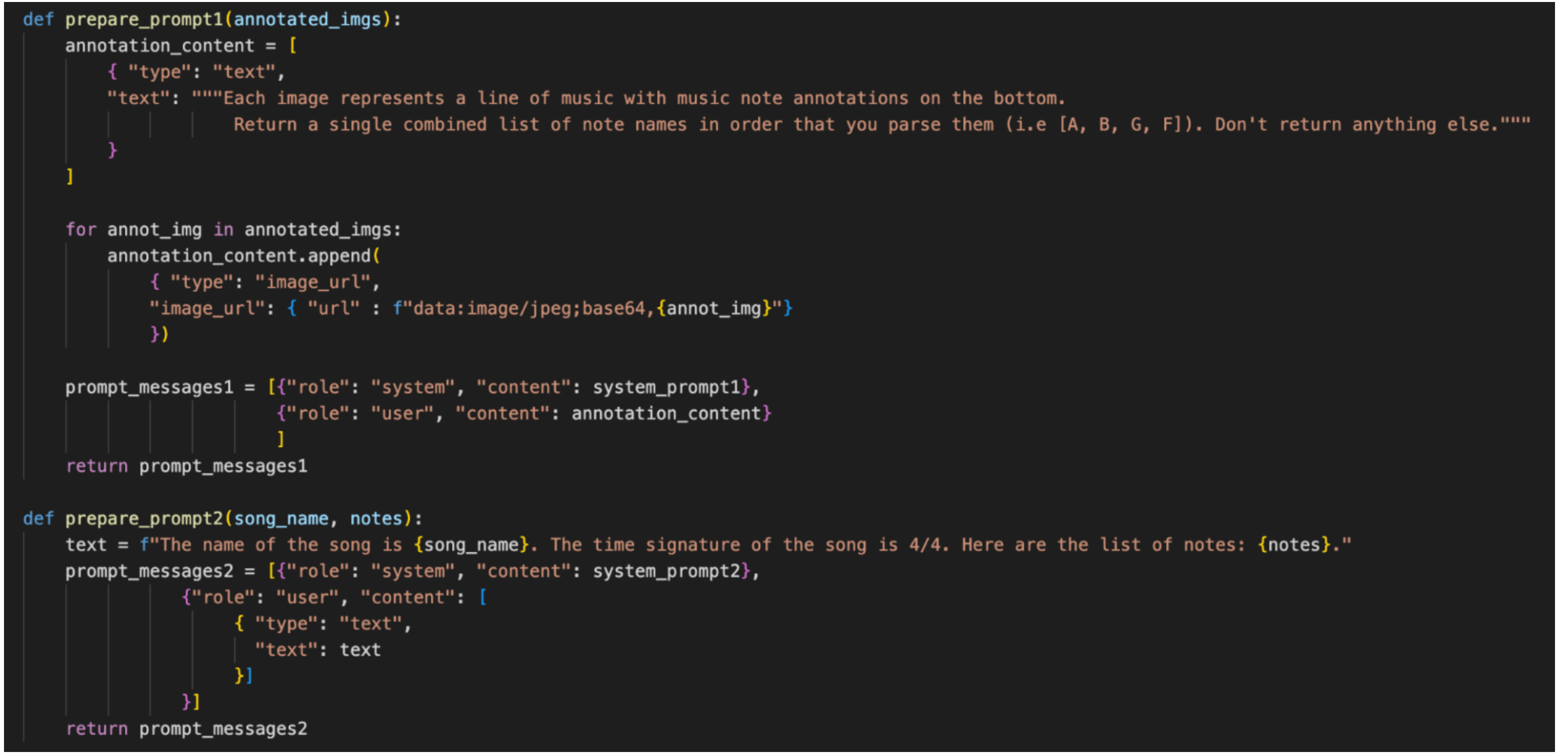
Agent specialized in determining the duration of notes based on examples.

Prompt Engineering



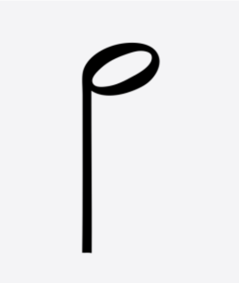
```
system_prompt1 = """
  You are an agent specialized in reading labelled music notes.
   Image Context:
  You will be given an image of a music sheet that contains some notes with a label below the note (a letter) indicating what note it is.
  Your goal is to return a list of note names in string format. Group each measure
    ['A', 'B', 'G', 'E', 'F', 'D', 'C', 'D']
   Don't include anything else in your response.
"""

system_prompt2 = """
    You are an agent specialized in composing music given the notes for a musical piece.

    You will be given the name of a song, the time signature, and the notes to the song in the form of a list
    (i.e ['A', 'B', 'G', 'E', 'F', 'D', 'C', 'D']). The possible note names in an octave are: C, D, E, F, G, A, B, C.

    Your goal is to assign a note duration to each note so that the note, duration combination will sound like the actual song provided.

    Return a list of tuples containing (note, duration) both string datatypes:
        (i.e [('C', 'Quarter'), ('C', 'Quarter'), ('G', 'Quarter'), ('G', 'Quarter'), ('A', 'Half')]).

    You can pick from the following list of durations: (Whole, Quarter, ThreeQuarter, Half, Eighth).
    Don't include anything else in your response.
"""
```

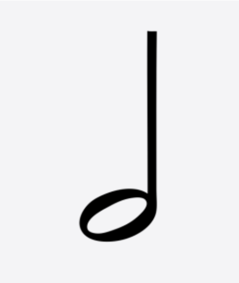
Prompt Engineering

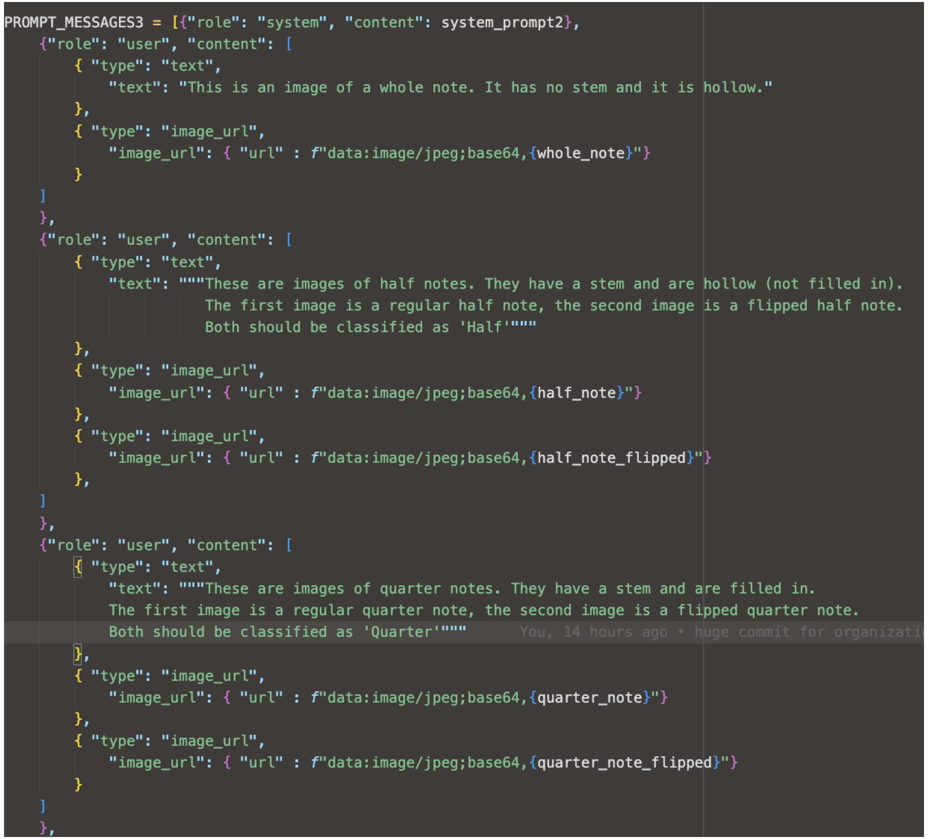


```python
def prepare_prompt1(annotated_imgs):
    annotation_content = [
        { "type": "text",
        "text": """Each image represents a line of music with music note annotations on the bottom.
                    Return a single combined list of note names in order that you parse them (i.e [A, B, G, F]). Don't return anything else."""
        }
    ]

    for annot_img in annotated_imgs:
        annotation_content.append(
            { "type": "image_url",
            "image_url": { "url" : f"data:image/jpeg;base64,{annot_img}"}
            })

    prompt_messages1 = [{"role": "system", "content": system_prompt1},
                        {"role": "user", "content": annotation_content}
                        ]
    return prompt_messages1

def prepare_prompt2(song_name, notes):
    text = f"The name of the song is {song_name}. The time signature of the song is 4/4. Here are the list of notes: {notes}."
    prompt_messages2 = [{"role": "system", "content": system_prompt2},
               {"role": "user", "content": [
                    { "type": "text",
                      "text": text
                    }]
               }]
    return prompt_messages2
```

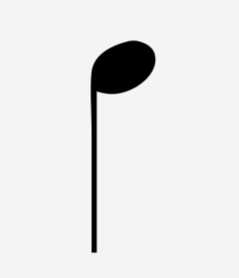
Accuracy Improvements

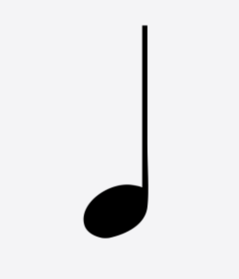






```
PROMPT_MESSAGES3 = [{"role": "system", "content": system_prompt2},
    {"role": "user", "content": [
        { "type": "text",
            "text": "This is an image of a whole note. It has no stem and it is hollow."
        },
        { "type": "image_url",
            "image_url": { "url" : f"data:image/jpeg;base64,{whole_note}"}
        }
    ]
    },
    {"role": "user", "content": [
        { "type": "text",
            "text": """These are images of half notes. They have a stem and are hollow (not filled in).
                       The first image is a regular half note, the second image is a flipped half note.
                       Both should be classified as 'Half'"""
        },
        { "type": "image_url",
            "image_url": { "url" : f"data:image/jpeg;base64,{half_note}"}
        },
        { "type": "image_url",
            "image_url": { "url" : f"data:image/jpeg;base64,{half_note_flipped}"}
        },
    ]
    },
    {"role": "user", "content": [
        { "type": "text",
            "text": """These are images of quarter notes. They have a stem and are filled in.
            The first image is a regular quarter note, the second image is a flipped quarter note.
            Both should be classified as 'Quarter'"""      You, 14 hours ago • huge commit for organizati
        },
        { "type": "image_url",
            "image_url": { "url" : f"data:image/jpeg;base64,{quarter_note}"}
        },
        { "type": "image_url",
            "image_url": { "url" : f"data:image/jpeg;base64,{quarter_note_flipped}"}
        }
    ]
    },
```





Sheet Music Reading Demo

Piano GUI Demo

What Worked

Prompt Engineering: was able to "teach" GPT what different notes looked like by providing it a library of annotated notes.

Camera Vision: successfully identified key elements (e.g. song titles, staff lines, bar lines).

Skew Correction: successfully reads sheet music that is off-skew.

What Didn't

True Optical Music Recognition (OMR): even after given context, GPT could not decipher notes completely correctly without labels.

- Could not read musical staffs with many measures and condensed notes.

Limitations of Cozmo's Camera: noisy images.

- Sheet music had to be created in MuseScore to be "clean."

Future Work

- Get OMR working without note annotations.
- Expand to sheet music with eighth and three quarter notes.
- Transpose sheet music that is not in the second octave into Cozmo's singable region.
- Transcribe sheet music in the bass clef.
- Integrate into one fsm – Cozmo reads sheet music and sings while "pressing" notes on the keyboard.

Technologies Used

- **Piano GUI:** https://github.com/plemaster01/PythonPiano
- MuseScore
- GPT-4 Vision API

THANK YOU!