Pitch Recognition and Note Detection from Sheet Music

Jeffrey Liang, Austin Liu, Seyoung Jang, Eric Wang

CSCI 1470 Spring 2025 Final Project

Introduction

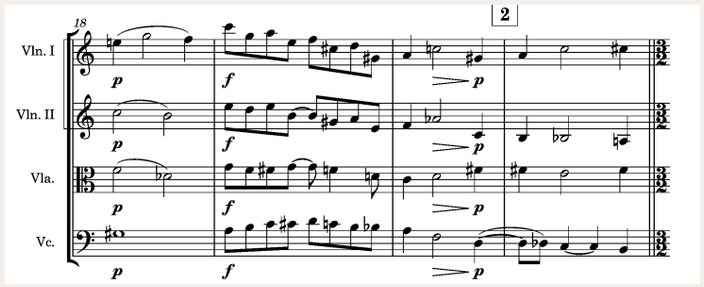
Our project takes PDF sheet music and will output an image with the pitches and type of notehead. Our pipeline extracts individual notes using **computer vision**, and outputs the pitches and types of notes in the music. We train a **convolutional neural network** to detect and localize musical notes from preprocessed images derived from the **DoReMI** dataset.

Our approach combines supervised learning, image segmentation, and CNN-based detection to extract relevant features from sheet music, which could eventually be used to automate the creation of OMR_XML files.

Data

From **DoReMi**, we used their scanned PDFs of beginner piano sheet music and songs, as well as their corresponding OMR (Optical Music Recognition) XML files.

The dataset contains **5,218 images** representing **44 songs**, where each song is divided into multiple pages. Each page has an associated **OMR_XML file.** Preprocessing was necessary to clean the labels, filter out irrelevant classes, and adjust bounding boxes to match cropped images focused solely on staff lines.



Results

Staff Line & Artifact Removal

- **Opening** kernel cleans up small noise and stray pixels (i.e. slurs, ties)
- **Closing** kernel used to fill in gaps in notes formed after the staff line removal

Vertical Blob Merging

- Thin vertical closing kernel to merge vertically adjacent components (i.e. parts of a time signature)

Custom Filter

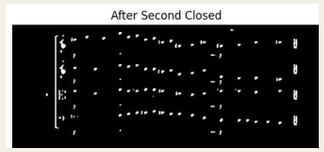
- **Area**: discard contours that are too small or too large to be noteheads
- **Aspect Ratio**: remove shapes that are too long or too narrow

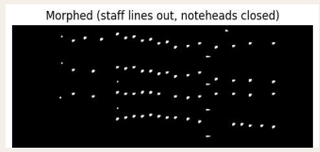
Image Segmentation

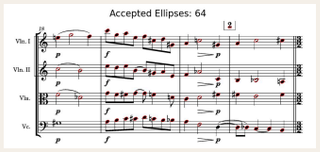
- Apply the **Watershed algorithm** to split remaining grouped blobs
- **Horizontal split**, cutting excessively tall blobs into notehead chunks before smoothing out edges
- **Series of tests:** Apply set of geometric filters to ensure each contour is ellipse-like in size, shape, and solidity — rejecting blobs that are too small, irregular, or poorly fit by an ellipse.

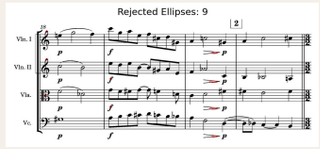
CNN Binary Classification

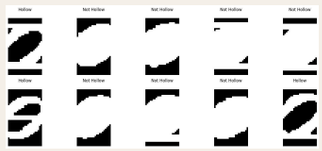
- Feed each cropped note into a trained CNN to classify it as hollow or filled note
- Outputs the final hollow vs. not hollow decision

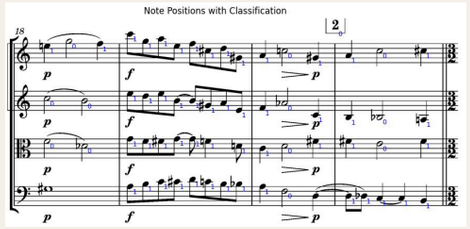










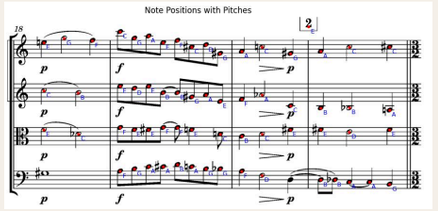


Notehead classification

- **0** hollow notehead → **whole/half notes**
- **1** non-hollow notehead → **quarter notes**

Pitch Identification

Map each note's vertical position to a pitch using the **staff's center and spacing.** Notes are compared against a lookup table that matches their **vertical offset.**



Takeaways/Reflections

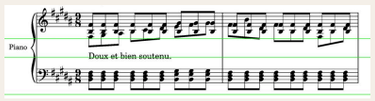
1. We learned that the model's performance is majorly dependent on the **quality of the input data**. We had to build our own pipeline from scratch -- **removing staff lines, extracting notehead regions, and tuning morphological operations**. A major challenge was balancing removal of unwanted noise without erasing noteheads. This involved carefully designing dilation, erosion, and closing steps, followed by rigorous contour and geometric filtering to retain only valid black, half, and whole notes.

2. Our initial goal was to train a multi-class classifier to distinguish black, half, and whole notes. But the shape similarities between hollow note types made it **difficult to separate half and whole notes**. As a solution, we switched to a **two-stage binary classification pipeline**: first detecting whether a note was filled (quarter/eighth/sixteenth) or hollow (half/whole), then distinguishing between half and whole notes using separate geometric heuristics and classifiers. This decomposition significantly improved our accuracy for detecting if its hollow or filled and we are still working on whole note classification.

Methods

1 Staff Line Extraction

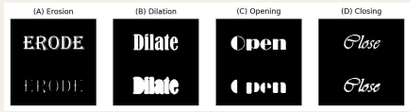
Using **binary inversion** and **horizontal projection** (summing across each row), we highlight staff lines as **peaks** and group every five lines into musical staves. We extract the upper bound, lower bound, and unit distance between each staff lines.



2 Morphological Operation

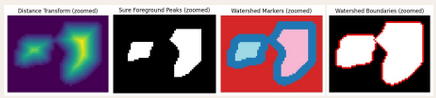
We applied morphological closing and opening operations to prepare the images for notehead extraction.

- **Closing** filled small gaps within broken noteheads, ensuring solid white blobs.
- **Opening** removed small specks and disconnected thin artifacts such as stems or slurs.



3 Split Blob

Despite morphological cleaning, closely spaced noteheads often appeared as merged blobs. To address this, we used a **watershed algorithm**. Watershed flooding **separated touching blobs into distinct regions.** Blobs with extreme aspect ratios were further refined by **horizontal splitting** to isolate vertically stacked notes.



4 CNN Classification

We trained a CNN to differentiate extracted notes into two classes: **hollow noteheads** and **non-hollow noteheads**.

5 Pitch Estimation

To determine the musical pitch of each detected notehead, we **compared each notehead's y-axis position to the staff line**. Using standard musical notation rules (e.g., notes sit either on lines or spaces), the vertical distance was mapped to a specific pitch.

Limitations/Challenges

1. **Staff-line removal:** Removing staff lines required extensive experimentation with morphological operations. At the same time, aggressive filtering risked removing parts of noteheads that overlapped with staff lines. We had to carefully **balance kernel size and iteration count** to eliminate lines while retaining musical symbols.

2. **Morphological filtering for note separation:** It was challenging to cleanly separate whole notes from surrounding noise/blobs. Our approach involved reconnecting broken outlines of hollow noteheads to make them more detectable**.** However, the same process sometimes unintentionally merged nearby notes or distorted the shape of filled notes. Finding the right balance between **restoring hollow noteheads and avoiding unintended merging** took significant trial and error.

3. **Note classification**: **Distinguishing whole vs. half notes was especially challenging**, as the only differences are subtle variations in size and ellipse orientation—features that are hard for a CNN to reliably learn and something we are still working on.