Optical Music Recognition from Music Sheet

CSE573: Computer Vision and Image Processing

Ninad Kale

September 30, 2022

---

1 Project Description

Recognise and play series of notes from given musical sheet

Playing music by reading musical sheets is difficult for beginners. I am playing piano for years now and I am still not able to read it properly. In this project, I want to solve this problem using computer vision. This problem statement is very similar to OCR, the only difference is instead of characters or numbers we are detecting certain notes and at what beat it is being played.

2 Strategy

The music sheet includes two bars, the top one for the right hand and the bottom one for the left, and notes are read from left to right. In this project, I tried to focus only on right-hand notes as it usually has a melody. First of all, I cropped all right-hand bars and then apply a customized algorithm on them for note detection.

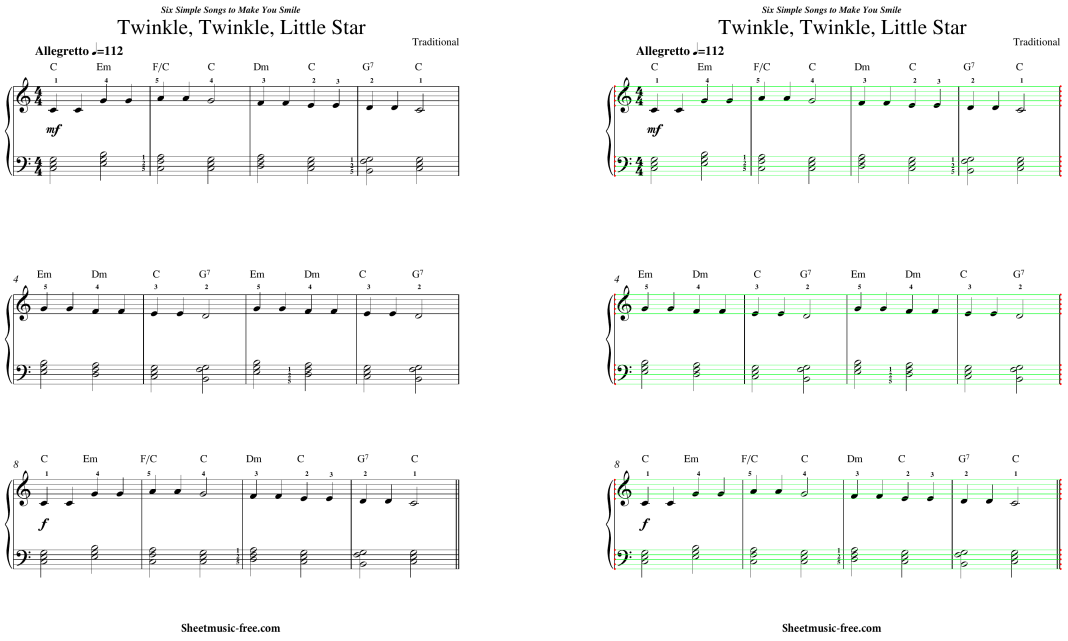
2.1 Region of interest detection

I used a musical sheet of the song Twinkle Twinkle Little Star. As it only contains two types of notes, half note, and full note, this breaks down the bigger problem to be solvable within the scope of this project.

First of all, the music sheet which is available in PDF is converted into PNG format
Steps to extract the right-hand bar in images:

- Apply horizontal filter to detect horizontal line
- Fit contour over the detected horizontal region to get corner points of each line
- Using point location, crop bar image

Kernel length has to be decided properly for it to only detect horizontal line bars and ignore others. Points given by contour have to be processed further to get points in the right format and remove unwanted points.



f

(a) Input image **(b)** Lines and corners detected

Figure 1: Prepossessing of Input Image

(a) Entire bar cropped **(b)** Right hand bar cropped

Figure 2: Extracting Bar Image

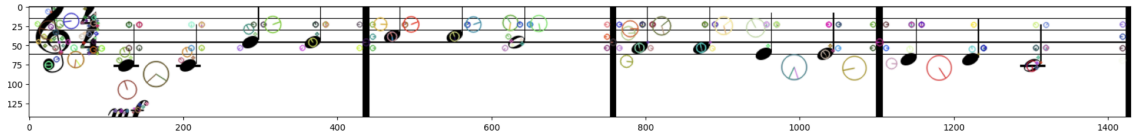
2.2 Note detection Algorithm

2.2.1 Template Matching

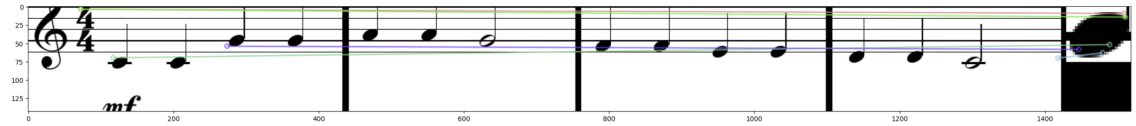
After discussing this problem with the professor, he suggested trying template matching. The issue with this approach is the localization of each note. Note is identified with the relative position of the blob with respect to horizontal lines. Also, I wanted to make a solution that is scalable and can be applied to any musical sheet. As the size of each sheet can vary, scale variant template matching will not be an optimal solution.

2.2.2 Sift Detector Matching

As SIFT is scale-invariant, I tested this approach too. To keep things simple I wanted to match the features of one clef/note with respect to the entire bar image. After passing images through SIFT key point detection, detected features were very bad for matching as can be shown below image. The reason behind this is music sheet is not feature-rich. Most of the image being binary is white with few shapes that too very small.



(a) SIFT features



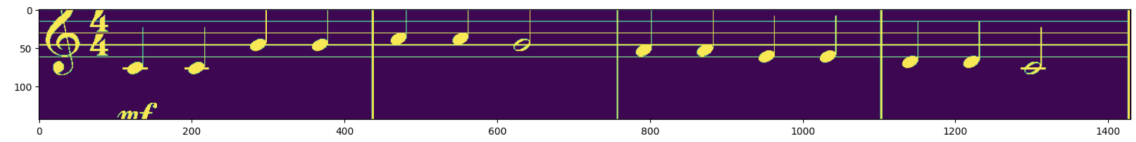
(b) Feature Mapping

Figure 3: SIFT

2.2.3 Customised Algorithm

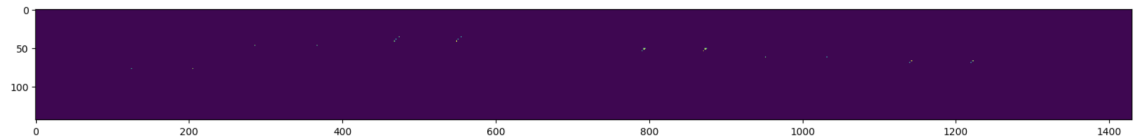
Finally, I tried to make my own algorithm based on concepts taught in class. If we observe the bar image, We already have horizontal line positions. The only thing we need is to detect notes and a method to differentiate between the full notes and half notes.
Algorithm:

- Erode image with a kernel of size 5. This will remove horizontal lines.
- Further, erode image with a kernel of size 7. This will help us to identify note location
- Dilate Image with a kernel of size 2 to get good localization of notes

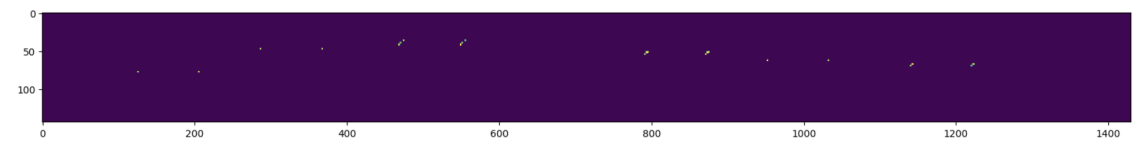


(a) Input Image

(b) Erode with kernel size of 5



(c) Erode with kernel size of 7



(d) Dilate with kernel size of 2

Key point to note here is we are only able to detect a half note. The full note is ignored. This is intentional as now we can make another set of filters in which we will detect full notes and thus we can have a way to differentiate between notes

- Iterating column-wise in an image, check if any of the point is high. If yes then it's a note. The distance between two identified notes should be greater than 50 pixels to avoid repetition.
- Apply a new set of filters to detect notes including full notes.

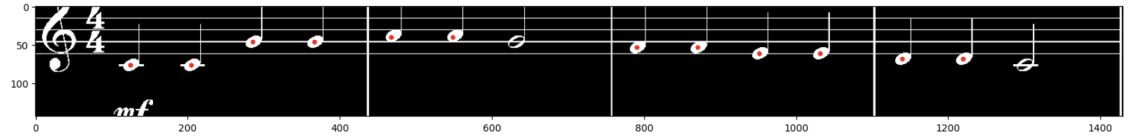
(e) Dilate Input Image

(f) Erode image

(g) Dilate image

(h) Erode image

With two different sets of filters, we get two different sets of images one with only half notes and one with all notes. If the note is only present in the former one then it's a half note else a full note



(i) Only half notes

(j) All notes

- Horizontal line position is used to identify each note. Accuracy of detection is 100%
- Musicalbeeps library in python is used to produce notes sound. Half note is played for 0.5sec and Full note for 1Sec

3 Comparing with state of art

- State of art Optical music recognition software can detect all kinds of notes. In our case, we can only detect full and half notes.
- I haven't taken into account tempo which will significantly affect the sound produced. As our input song is simple it is not creating a huge difference.
- Left-hand notations are ignored in our algorithm which is included in the state of the art algorithm.