Credit Hours System

CMPN446-

Image Processing & Computer Vision

Cairo University

Faculty of Engineering

Sheet Music Reader

Team 16

Submitted by:

Huda Sherif	1164373
Emad Alaaeldin	1162431
Aly Ahmed	1165430
Mohamed Osama	1162367

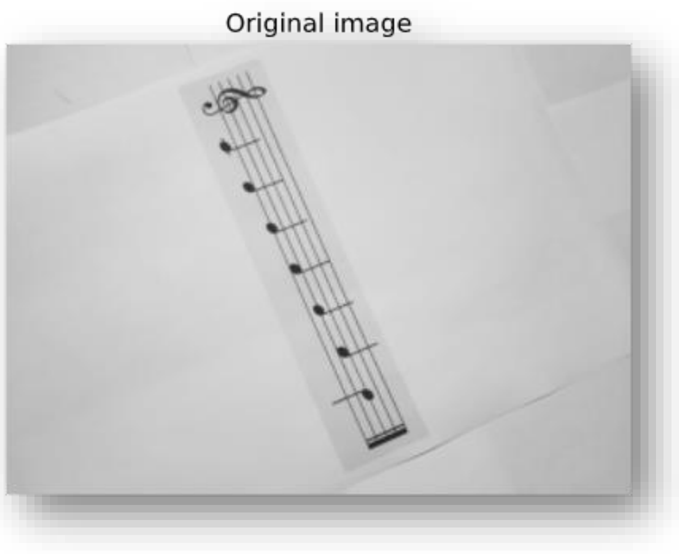
Used algorithms:

A high-level overview on the pipeline of our work:

1- Binarization
2- De-skewing
3- Staff Line detection and removal
4- Segmenting the image in to staff collections
5- Segmenting each collection into symbols
6- Feature Extraction & Classification

Explanation of each level in details:

In our explanation, we will use the below input image to demonstrate each part (27.jpg)

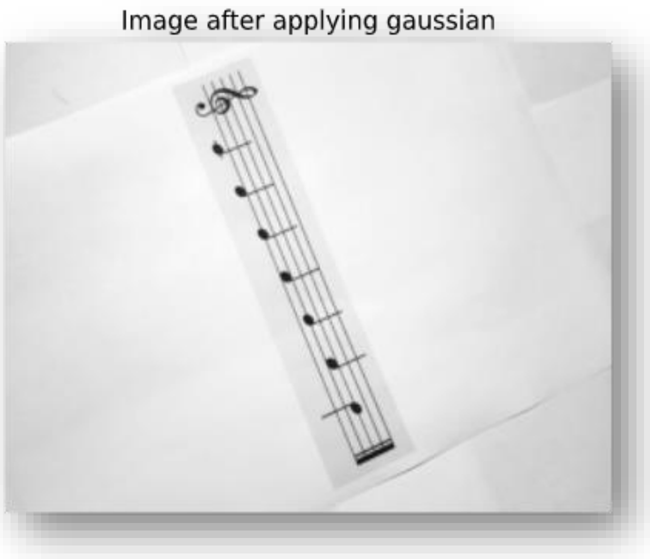


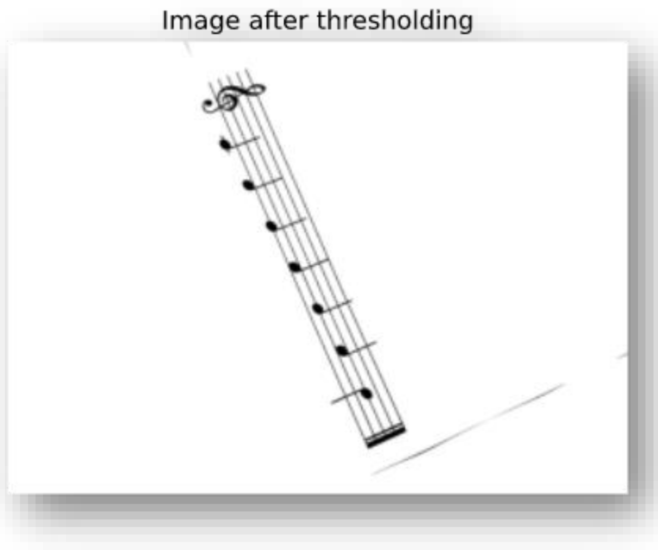
1- **Binarization**:

We noticed that each image had different back ground colors and different gray levels thus we could not use a default threshold to binarize the image as this would have led to loss of important details in many of the images and very low performance.

Otsu and adaptive thresholding also caused very low performance and caused loss in many important details so we had to try a different approach

We were able to find a very good algorithm that starts with a Gaussian filter followed by iterations of otsu algorithm till divergence and this approach led to very good binarization and most details were protected





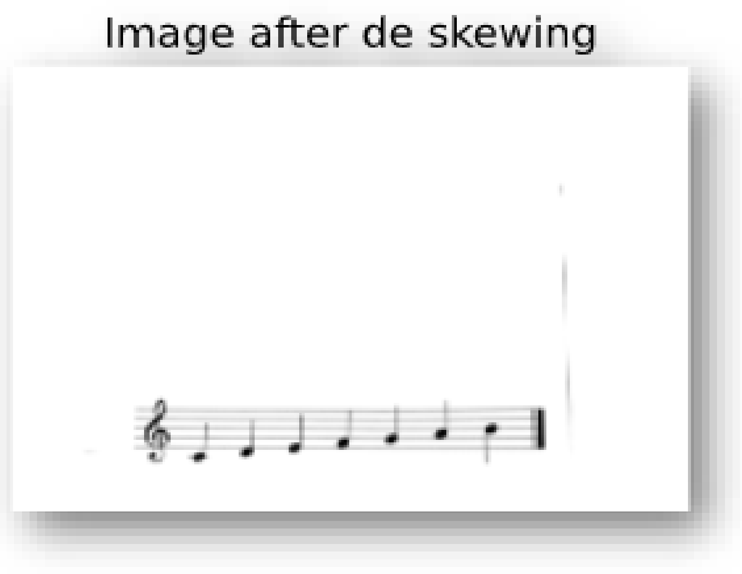
2- **De-skewing**:

For de-skewing, we used Hough transform to get the most common line in the image and find the angle of the line and rotate and the picture according to this angle

However, during this phase we faced many problems, it is whether to rotate the image clockwise or anti clockwise, as some of the pictures after rotation ended up upside down and this caused our model to have 0 performance as the picture is processed in a wrong way, thus after researching some papers about this issue, we found out that with OMR documents this issue is very common as there is no accurate way to determine if the document is in its correct orientations or is it inverted, however a very common approach for this issue is studying the dataset and studying the angles of each document that cause them to de-skew in a right way, this approach takes the assumption that data is usually entered in the same way as the data set.

After following this approach, we followed this strategy that if the angle resulting from the algorithm is positive, we rotate with this angle if not we will add 180 to the angle of rotation.

This approach led to very fair results; it was not very accurate but it gave much better performance.



3- Staff Line detection and removal:

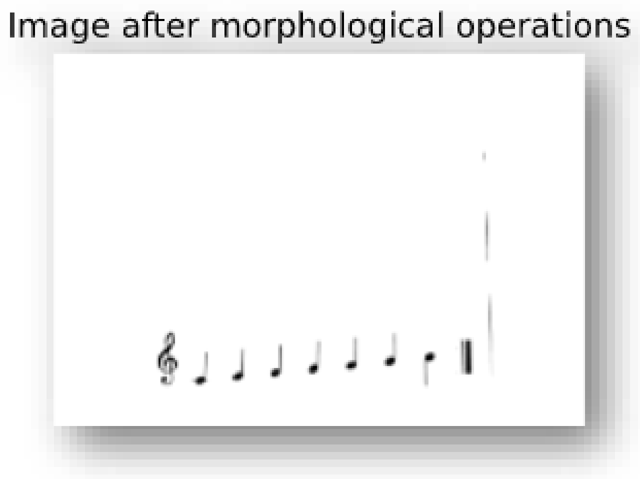
First, we started by applying RLE and studying the black pixel runs and white pixel runs in each column, and then using this data we were able to get the most common black run sizes which is our staff thickened and the most common white run sizes which is out staff space

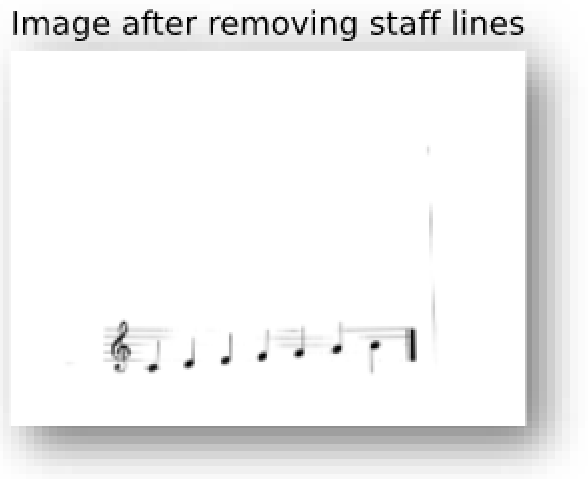
After that we applied an algorithm very similar to RLE, however it studies rows instead of columns and it is used to approximate the candidate rows where the staff lines are found by finding the rows with black pixels a certain percentage of black pixels from the width of the picture. We had to adapt it in some way with a threshold not to move in a straight path and staff lines do not always follow a straight line and some are discontinued.

With this we were able find the positions of our staff lines and divide our staff lines into collections of 5 according to the estimated staff thickness and space

Then we remove the black runs we got from the picture on two conditions, that they are present at the staff positions and that their size s around the size of the estimated thickness of the staff line.

After this step we apply morphological operations to remove remaining staff lines that were not detected by the previous algorithm because as mentioned before staff lines are sometimes not parallel and sometimes are skewed and don’t follow a straight path.



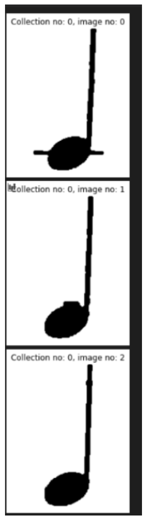


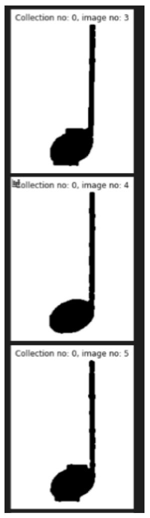
4- <u>Segmenting the image in to staff collections:</u>

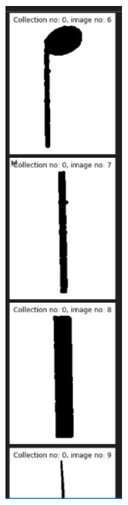
We are then able to segment our image into smaller images each containing the symbols of a single staff collection, this is done by dividing our staff lines into fives each represent a collection.

5- <u>Segmenting each collection into symbols:</u>

We then take each collection and segment them in to symbols. This is done by getting the contours and then the bounding boxes, with cutoff size/area of 80 to remove unwanted noise and return all the segmented images as a list.


Collection no: 0, image no: 0
Collection no: 0, image no: 1
Collection no: 0, image no: 2


Collection no: 0, image no: 3
Collection no: 0, image no: 4
Collection no: 0, image no: 5


Collection no: 0, image no: 6
Collection no: 0, image no: 7
Collection no: 0, image no: 8
Collection no: 0, image no: 9

6- <u>Removing starting symbols:</u>

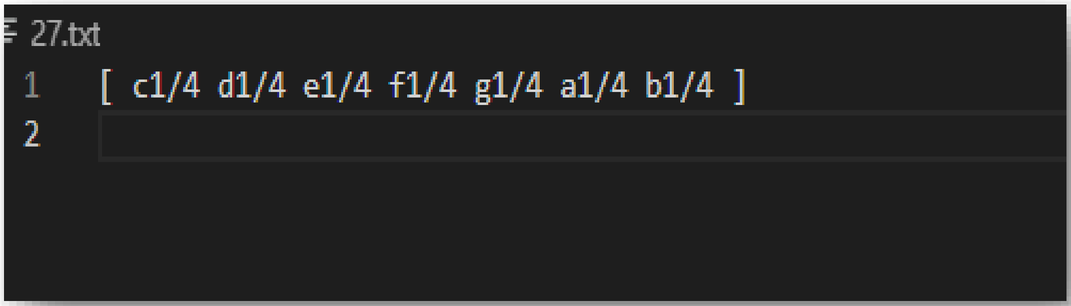
We now have the images segmented as a list. We the remove the first unwanted symbols and noise, and then check if there are numbers to classify. We do that by checking the y and x positions, and then classify the numbers by taking the lower right of the image and checking if there is a vertical line, it will be ‘4’ otherwise it will be ‘2’.

7- <u>Classification of symbols:</u>

We now have the symbols to be classified. We use a structured if condition and with each condition we eliminate symbols until we reach the desired output.

We first check if it is an accidental or regular symbol, which is done by checking the height, if it is greater than 3.5*staff space, then it is a regular symbol, otherwise it is an accidental. For regular symbols, we count the number of black heads, if there are no black heads, then we count the white heads and if the is a white head, then we know it is a hole with stem, however if we found 1 black head, it is either a note or flags, so we count the number of flags, if we found flags it will be classified as flag, if not it will be classified as regular note. If there was more than 1 black head, then we know it is either a chord or a beam, so we count the number of vertical lines/stems, if there is only one vertical line/stem, then we know it is a chord and we see the position of each black hole, if there was more than 1 vertical line/stem, then we know it is a beam, so we see the position of each black hole and return the desired string accordingly.

For accidentals, we first get the percentage of black pixels to white pixels, if it is more than 70% then we know it is a dot (‘.’), then, as we count the number of vertical lines, if it is 1, then we know it is a b, if it is more than 1, then we know it is either a bb, # or rectangular shape, so we see the number of vertical lines in the upper 30% of the image, if there is one vertical line, then it is the rectangular shape, if it is more than 1 then we check the number of horizontal lines in the upper 30% of the image, if there is a horizontal line then it is the #, otherwise it is a bb. Now, we have the cross (‘x’) and the hole (‘o’) remaining, for this we use hough transform and get the highest two angles, if they are around 45 and 135 degrees we know it is a cross, otherwise it is the hole.



```
[ c1/4 d1/4 e1/4 f1/4 g1/4 a1/4 b1/4 ]
```

Work division between team members:

Name	Workload
Emad Alaa	1. Image preprocessing: staff line removal and detecting their positions 2. Feature Extraction and classification
Aly Ahmed	1. Segmentation of collections (collection= 5 staff lines) 2. Feature Extraction and classification 3. Docker setup
Mohamed Osama	1. Image preprocessing: Thresholding, Noise removal, deskweing 2. Semantic Reconstruction
Huda Sherif	1. Segmenting each separate collection 2. Feature Extraction and classification

Accuracy and performance:

Accuracy over digital images ~ 95%

Accuracy over camera captured ~ 35 %

Conclusion and references:

- https://www.researchgate.net/publication/220163392_Optical_recognition_of_music_symbols_-_A_comparative_study
- https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiP5aCxzpTuAhXpaRUIHTHpCjQQFjAAegQIARAC&url=https%3A%2F%2Fwww.researchgate.net%2Fpublication%2F221050952_Staff_Line_Detection_and_Removal_with_Stable_Paths&usg=AOvVaw2aKq177BTDg7gPo0KR3gAu
- https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiP5aCxzpTuAhXpaRUIHTHpCjQQFjABegQIAhAC&url=https%3A%2F%2Fwww.sciencepubco.com%2Findex.php%2FJACST%2Farticle%2Fview%2F3196&usg=AOvVaw1zmur_K2T5Pnv3nHdiEDAc
- https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjnl6uWz5TuAhWERxUIHfyNC9sQFjAAegQIAhAC&url=https%3A%2F%2Fwww.researchgate.net%2Fpublication%2F3939804_Optical_music_sheet_segmentation&usg=AOvVaw1t236DfIOFsN6KdJPY9Zve
- https://link.springer.com/article/10.1007/s11042-017-5169-9
- https://web.stanford.edu/class/ee368/Project_Spring_1415/Reports/Verma_Harris.pdf

- G. Read. Music Notation: A Manual of Modern Practice. Taplinger, New York, 2 edition, 1969. ISBN 0-8008-5459-4.

- L. J. Tard´on, S. Sammartino, I. Barbancho, V. G´omez, and A.Oliver. Optical music recognition for scores written in white mensural notation. EURASIP Journal on Image and Video Processing, 2009:6:3–6:3, February 2009. ISSN 1687-5176.

- K. T. Reed and J. R. Parker. Automatic computer recognition of printed music. In Proceedings of the 13th International Conference on Pattern Recognition, volume 3, pages 803–807, Aug. 1996.

- Optical Music Recognition - State-of-the-Art and Open Issues

- Metric Learning for Music Symbol Recognition

- I. Fujinaga, “Staff detection and removal,” in Visual Perception of Music Notation: On-Line and Off-Line Recognition, S. George, Ed. Idea Group Inc., 2004, pp. 1–39.