Optical Music Recognition

Sakshamdeep Singh
50475857
ssingh86@buffalo.edu

Shashank Vivek Baratwaj
50479686
svivekba@buffalo.edu

1 Overview of project

1.1 Abstract

In this project we incorporated a state of the art deep learning YOLOv8 model and OpenCV operations to build a system that reads musical sheets and plays it. We trained the YOLO model using different types of notes such as whole, half, and quarter notes.

A significant challenge in detecting musical notes from sheets is to determine the duration of each note. This requires identifying the particular staff line on which the note lies and determining the note's length based on its position in relation to the other notes and the time signature of the music.

The output of the system was the detected notes on the musical sheets including the position of the notes on the staff line to indicate the perfect pitch of the respective notes. Overall, the report presents a novel approach to using YOLO and OpenCV for music note detection, with potential applications in music transcription and analysis.

Additionally, the paper describes that after the YOLO model detects the musical notes on the sheet, the system plays the corresponding notes. This is achieved by converting the position of the detected notes on the staff line to their respective pitches and using a MIDI library to play the notes. Thus, the system not only detects the musical notes but also produces an audible output of the sheet's contents, making it a useful tool for musicians and music educators.

1.2 Related Works

Related works in the field of music transcription and analysis have focused on using various techniques to automatically detect and transcribe musical notation from scanned images or audio signals. These techniques range from deep learning approaches that extract fundamental frequencies from audio signals to hybrid models that combine convolutional neural networks with hidden Markov models for sequence modeling.

Other works have focused on optical music recognition, including staff line detection, note recognition, and score alignment. Overall, these related works emphasize the importance of using computer vision and machine learning techniques to improve music education and performance.

1.3 Contributions

Our project introduces an innovative approach by utilizing the advanced YOLOv8 model, the latest addition to the YOLO family, for note detection. Instead of relying on a sequence model, we employ basic programming techniques to obtain the sequence. Furthermore, we leverage basic OpenCV operations to detect staff lines, and ultimately, the sequence is played using the pygame library.

2 Motivation

The motivation behind the project of musical note recognition using YOLO object detection and OpenCV for staff line identification is to develop a system that can automate the process of music transcription, which is the process of converting sheet music into a digital format. Transcribing music can be a time-consuming and error-prone task, requiring significant skill and training. By developing an automated system for music transcription, musicians and music educators can save time and effort, while also ensuring greater accuracy and consistency in their transcriptions.

Additionally, the proposed system can enable new applications in the field of music technology. For example, the system can be used to create interactive music apps or games that allow users to play along with sheet music in real-time. The system can also be used to analyze and extract musical patterns and structures from large collections of sheet music, facilitating research and analysis in musicology and music theory.

3 Approach

3.1 Algorithm

We obtained sample piano notes from a website, and annotated them using the labelImg tool, saving the annotations in YOLO format. With this dataset, we trained the state of the art YOLOv8 model to detect musical notes. Once trained, the model was applied to an input image to detect notes.

The subsequent task was to locate the position of the notes on the staff lines. To achieve this, we utilized horizontal structuring elements in morphological operations to identify the staff lines. To obtain the centroids of the note blobs and calculate their distance from the staff lines, we inverted the image to prevent errors in binary image processing. Horizontal and vertical lines were then eliminated to isolate the notes as the primary object of interest. The steps involved in finding the centroids of the notes are outlined below:

1. The image was thresholded using Otsu and inverse binary thresholding to separate the object (notes) from the background.
2. The contours of the binary image were found using cv.CHAIN_APPROX_SIMPLE, which removes redundant points and compresses the contour to save memory.
3. Once the contours of the object were obtained, the centroid was found using the moments of the image. The moments of an image are mathematical descriptors that can be used to calculate various properties of the object, including its centroid.

Overall, this approach was effective in detecting the position of musical notes on the staff lines, which allowed for further analysis of their kind and pitch.The centroids of the note blobs were used to calculate the distance between each note and the staff line to determine the pitch. Once the pitch of each note was determined, they were played back using MIDI and Pygame libraries.

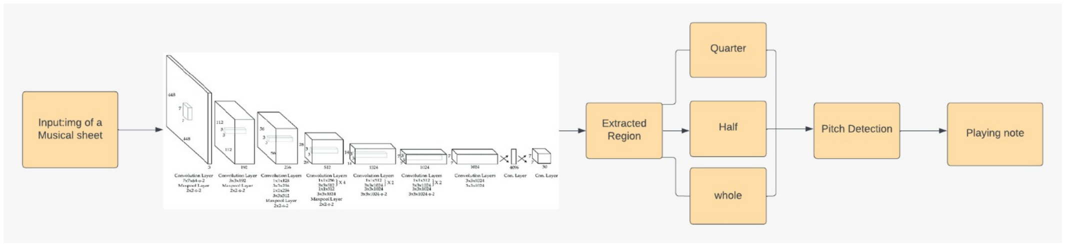


Figure 1: Methodology

3.2 Pros and Cons of our algorithm

Our methodology utilizes the YOLO model for object detection and employs cv2 operations for subsequent processing. This approach is well-suited for applications that demand quick and precise detection. However, while our approach outperforms non-deep learning methodologies, it exhibits suboptimal performance when applied to intricate notes.

3.3 Aspects that we have coded

We incorporated a pre-trained YOLO model from Ultralytics and customized it to fit our needs, tailoring it specifically for our application. Although we utilized several pre-existing functions, we coded the entire solution from scratch. We made sure to properly cite all the libraries that were used at the bottom of our work.

4 Experimental Protocol

4.1 Dataset

Our yolo model was trained on a dataset that we annotated, we chose 8 music sheets containing 17 songs in total.

`https://www.capotastomusic.com/piano-sheet-music/basic.htm`

We manually annotated the notes in the music sheets for training purposes using LabelImg. We obtained around 500 instances which we split into training,validation and testing in the ratio 5:2:1 respectively. Since our project's key idea was identification of notes, our dataset was extremely vital in the implementation of our project. Our project's evaluation metrics include precision, recall and f1 score.

4.2 Evaluation Metrics

- **Precision**: Precision measures the accuracy of the model's positive predictions. It is the ratio of the number of true positive detections to the total number of detections made by the model. In other words, precision is the fraction of positive detections that are actually positive.
- **Recall**: Recall measures the model's ability to detect all positive instances. It is the ratio of the number of true positive detections to the total number of actual positive instances in the test set. In other words, recall is the fraction of actual positive instances that are correctly detected by the model.
- **mAP**: mAP is a summary metric that measures the average precision across different recall levels. It is calculated by computing the precision at different recall levels, and then taking the mean over those precision values. mAP is typically reported at different intersection over union (IoU) thresholds, such as IoU=0.5 and IoU=0.5:0.95.

In summary, precision measures the accuracy of positive detections, recall measures the ability of the model to detect all positive instances, and mAP summarizes the average precision across different recall levels, typically at different IoU thresholds.

4.3 Software and Hardware Requirements

The project implementation relied on the following hardware and software specifications:

1. **Processor**: 12th Gen Intel(R) Core(TM) i7-12700H, with a base clock speed of 2.30 GHz.
2. **Memory**: 16.0 GB of installed RAM, with 15.7 GB usable.
3. **Operating System**: 64-bit version with x64-based processor.
4. **Graphics**: NVIDIA GeForce RTX 3060 GPU, with 6GB dedicated video memory.

5 Results

The performance of the custom-trained YOLOv8 model was evaluated on the training, validation and test sets. The results are presented below:

5.1 Training Set Evaluation

Class	Instances	Box(Precision)	Recall	mAP50	mAP50-95
all	261	0.879	0.93	0.974	0.5
half	106	0.714	0.991	0.975	0.487
quarter	124	0.961	1	0.995	0.525
whole	31	0.961	0.799	0.951	0.487

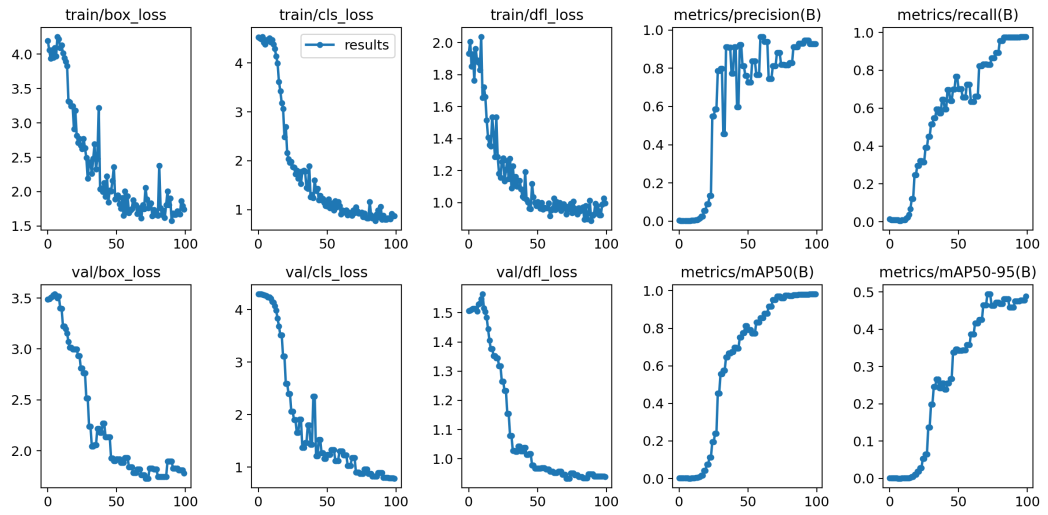


Figure 2: Training Graphs

5.2 Validation Set Evaluation

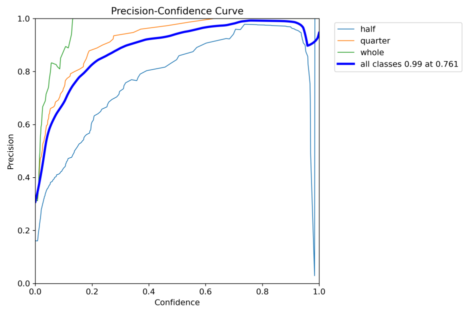
Class	Instances	Box(Precision)	Recall	mAP50	mAP50-95
all	144	0.881	0.83	0.97	0.495
half	51	0.706	0.98	0.959	0.476
quarter	73	0.937	0.986	0.987	0.479
whole	20	1	0.524	0.965	0.53

5.3 Test Set Evaluation

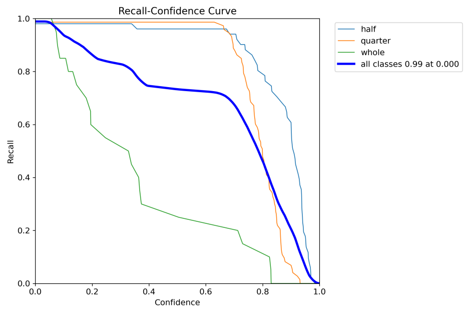
Class	Instances	Box(Precision)	Recall	mAP50	mAP50-95
all	58	0.876	0.906	0.986	0.529
half	25	0.694	1	0.968	0.52
quarter	18	0.933	1	0.995	0.499
whole	15	1	0.719	0.995	0.566

6 Analysis

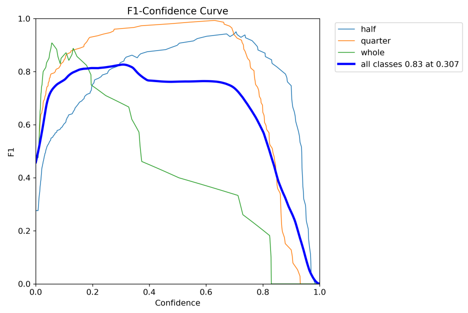
From the above results our findings showcase the success of the custom-trained YOLOv8 model in detecting musical notes. The high precision, recall, and mAP50 scores on both the training and validation sets, along with the promising performance in staff line detection, suggest that our system has the potential to enhance the current optical musical recognition performance.



(a) Precision Confidence Curve



(b) Recall Confidence Curve



(c) F1 Curve

Figure 3: Evalution Metric Graphs

7 Discussion and Lessons Learned

Through this project, we have learned several key lessons about computer vision, machine learning, and music processing. Specifically, we have learned about:

1. **Object detection**: We have gained hands-on experience with object detection using the YOLOv8 model. We have learned how to train the model on a custom dataset and how to fine-tune the model for better performance.
2. **Image processing**: We have learned how to use OpenCV functions to process musical sheet images to detect the lines and make them more suitable for object detection.
3. **MIDI and audio processing**: We have gained experience in using the midiutil and Pygame libraries to generate MIDI files and play them as audio.
4. **Cross-disciplinary skills**: We have learned how to integrate skills from different fields, such as computer vision, machine learning, and music processing, to create a meaningful and functional project.

In the future, we hope to apply these skills and lessons learned to other projects, such as music transcription, audio synthesis, and sound recognition. Additionally, we believe that this project has taught me the importance of creativity, perseverance, and teamwork in solving complex problems. These skills will be invaluable as we pursue further research and development in the field of AI and computer science.

8 Bibliography

[1] Huang, Z.; Jia, X.; Guo, Y. State-of-the-Art Model for Music Object Recognition with Deep Learning. Appl. Sci. 2019, 9, 2645. https://doi.org/10.3390/app9132645

[2] Chan Li, "A Deep Learning-Based Piano Music Notation Recognition Method", Computational Intelligence and Neuroscience, vol. 2022, Article ID 2278683, 9 pages, 2022. https://doi.org/10.1155/2022/2278683

3. https://docs.opencv.org/4.x/index.html

4. https://ultralytics.com/

5. https://github.com/ultralytics/ultralytics

6. https://pytorch.org/docs/stable/index.html