Solving a complex Music Note Detector

Artificial Vision | Universidad de las Américas Puebla

Introduction

This project addresses the challenge of automatically interpreting sheet music. The core issue possesses a few challenging characteristics:

- **System must accurately detect and classify a wide variety of visually similar symbols with high precision. Yet, musical notes are small within a large sheet music image, making it difficult for standard object detection models to locate reliably.**
- **Publicly available datasets (like DeepScores V2) are very much unbalanced.**
- **Real world sheet music can have poor contrast, background noise and artifacts that require robust preprocessing.**

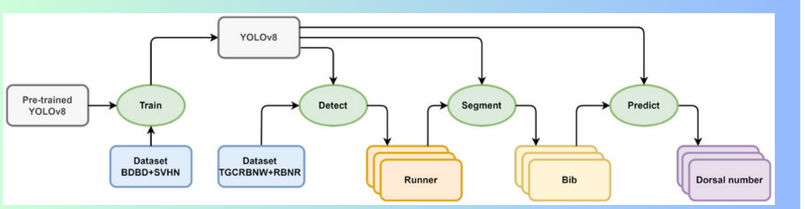
Proposed model

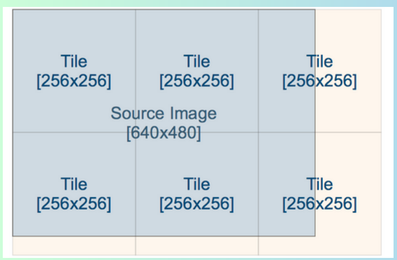
Core model:

- **YOLOv8: selected for its optimal balance of high speed and accuracy, perfect for real-time object detection. Used to detect and classify multiple note classes.**

Preprocessing pipeline:

- **CLAHE: image processing technique to increase local contrast and enhance edges of music symbols.**
- **Tiling: input image is split into four tiles to address the small object detection challenge.**





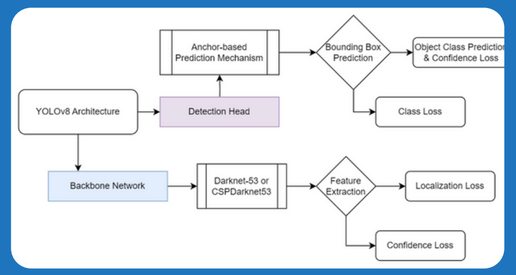
$$cdf(k) = \sum_{i=0}^{k} p(i)$$

Figs. 1,2,3: YOLOv8, Tiling, and CLAHE respectively

Proposed model - diagram

Final test included the mentioned preprocessing methods as well as YOLOv8:

1. Map classes from DeepScore dataset
2. Create dataset stucture
3. Convert with more data, tiling and preprocessing
4. Train with YOLOv8s



5. Evaluate

Fig. 4 - proposed final approach (diagram)

Applied resources & tools

- **Software / Libraries: Python, PyTorch, Ultralytics, YOLOv8, OpenCV**
- **Techniques: Tiling, CLAHE, dataset augmentation.**
- **Hardware: NVIDIA GPU (Google Colab's 4T GPU).**

Analysis of solutions

Model Version	Baseline YOLOv8	YOLOv8 + preprocessing
Accuracy (mAP50-95)	31.33%	26.70%
Precision	86.88%	74.70%
Recall	35.04%	31.90%
F1-score	47.82%	44.70%

Table 1 - comparison of metrics from two best-performing models

Results

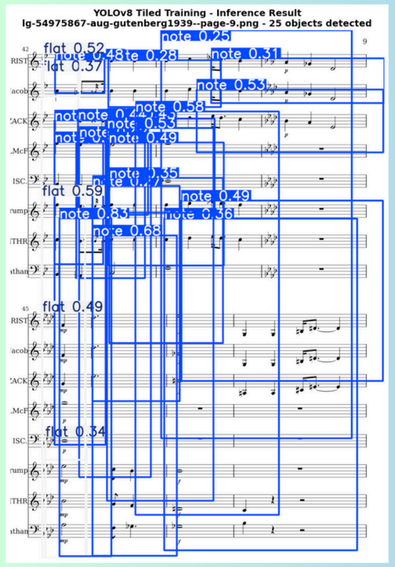


Fig. 5 - note detection from final proposed model

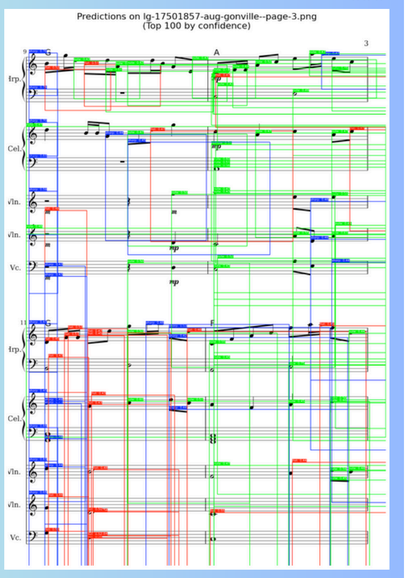


Fig. 6 - note detection from YOLOv8-only model

Contrary to our hypothesis, the preprocessing pipeline resulten in a lower accuracy (mAP) and slower inference time.
This might be because CLAHE can introduce unnatural artifacts that confuse the neural network. Also, Tiling may have amplified the dataset's inherent class imbalance.

Conclusions

The comparative results between YOLOv8 and YOLOv8 with preprocessing show that the baseline model performs better across all metrics. This outcome demonstrates that complex preprocessing pipelines do not guarantee performance gains and may even hinder learning when not precisely tuned.

However, the evaluation provides valuable insight into the actual "bump" of the task: class imbalance. Future work will prioritize solutions such as advanced augmentation and resampling methods, which are likely to output more impactful improvements than additional image preprocessing.