FACULTY OF ENGINEERING AND ARCHITECTURE

Feature-Based Detection of Handwritten Annotations on Historical Sheet Music

Ossama Belliraj
Student number: 02103969

Supervisors: Prof. dr. ir. Jan Aelterman, dhr. Nicholas Cornia (FAAM)
Counsellor: Anoek Strumane

Master's dissertation submitted in order to obtain the degree of Master of Science in Information Engineering Technology

Academic year 2024-2025

GHENT UNIVERSITY

Acknowledgment

This thesis has easily been the hardest challenge I have been dealt with in my academic career. Many times have I considered leaving this behind as it stood in front of me like a mountain that not many are able to traverse. But thanks to the many lovely people in my life, I can say that I was able to bring this hurdle to a satisfactory end.

I would like to thank my friends who have always stood behind me as I many times became unsure and hopeless in front of this task. I would like the thank FAAM and Nicholas Cornia who have put this challenge in front of me, believing I could tackle this despite my many failed attempts. I would like to thank Professor Aelterman who allowed me some breathing room when the deadline was near. I would like to thank my Mother who always supported me with a humorous tone when I could not get a smile across my face. And last but not least I would like to thank my counselor, Anoek Strumane, for always helping me find the right direction when I felt lost, and always pushing when it was my motivation who was lost.

Abstract

Feature-Based Detection of Handwritten Annotations on Historical Sheet Music

Ossama Belliraj

Student number:02103969

Supervisors: Prof. dr. ir. Jan Aelterman, dhr. Nicholas Cornia (FAAM)

Counsellor: Anoek Strumane

Academic year 2024-2025

The digitization of cultural archives has changed research in the humanities, yet the automated analysis of complex historical documents remains a challenge. Historical music scores can contain not only the printed content but also the valuable handwritten annotations from composers and performers that offer deep insight into historical performances. Optical Music Recognition tools are typically trained on clean, modern music scores that fail when confronted with the challenges brought by historical music scores. This thesis tries to address these challenges by designing, implementing, and evaluating a system to accurately differentiate between handwritten annotations and the printed content. The proposed methodology employs a multi-stage pipeline comprised of a binarization process for degraded documents, feature engineering based on shape, texture, color, and edge descriptors, and classification using a Support Vector Machine. The system is evaluated on the Flemish Archive for Annotated Music dataset, which contains 526 images with approximately 10000 annotations. The SVM model uses a Radial Basis Function (RBF) kernel that achieves a test set accuracy of 90.52% and an ROC AUC of 0.96. However a crucial limitation of this research is the imprecision of the ground truth data. The bounding boxes that indicate the location of annotations often include background pixels and pixels from printed elements, meaning the high accuracy should be interpreted with caution. This work does successfully establish a baseline for future research and demonstrates a viable path toward solving the challenging problem of analyzing annotated historical music scores.

Keywords: Optical Music Recognition (OMR), Document Image Analysis, Feature engineering, Support Vector Machine (SVM)

Feature-Based Detection of Handwritten Annotations on Historical Sheet Music

Ossama Belliraj

Supervisor(s): Prof. dr. ir. Jan Aelterman, dhr. Nicholas Cornia (FAAM)

***Abstract*—This paper proposes a robust, automated process designed to differentiate handwritten annotations from the original printed content in digitized music scores. The proposed methodology employs a multi-stage pipeline that includes a binarization process for degraded documents, comprehensive feature engineering and classification using Support Vector Machine (SVM).**

***Keywords*— Optical Music recognition (OMR), Document Image Analysis, Handwritten/Printed Text Separation**

I. Introduction

The digitization of historical documents has transformed research in the humanities, but automated analysis remains a challenge. Historical music scores are complex artifacts containing not only printed music but also a layer of handwritten annotations that offer invaluable insight into historical performances [6]. Optical Music Recognition (OMR) tools are usually designed for clean, modern music scores that often fail to process these annotations, creating a bottleneck for large-scale musicological studies [1],[2]. The Flemish Archive for Annotated Music (FAAM) offers a dataset containing scanned copies of historical music scores with a Json file containing information about the location of annotations on these music scores.

II. DataSet

The Flemish Archive for Annotated Music dataset consists of 526 RGB images of digitized historical music scores. Together with that comes a COCO JSON file that holds information about the location of approximately 10000 annotations across the 526 images. The annotations are indicated on the images using bounding boxes that show the approximate location of the annotations.

The most critical challenge that arises from this dataset is imprecision of the ground truth. The bounding boxes do not only contain pixels pertaining to annotations but always have background pixels and sometimes have pixels of printed content. This makes the evaluation of proper segmentation between handwritten annotations and printed content impossible. This subsequently makes labeling the printed content as label 0 and the handwritten annotations as label 1 very tricky.

III. Methodology

The proposed system follows a three-stage process: (1) Preprocessing, (2) Feature Extraction, and (3) Model Training and Classification.

A. Preprocessing

A robust preprocessing stage is crucial given the degraded nature of historical documents. A multi-stage binarization process is proposed to handle challenges such as contrast variation, faint annotations and uneven illumination. Standard Binarization techniques proved insufficient. Global thresholding techniques such as Otsu's thresholding tend to miss faint annotations due to their low contrast compared to printed elements [4]. Local adaptive thresholding techniques introduce significant noise in areas that are mainly background. The proposed approach combines their strengths and adds other image processing techniques to alleviate the weaknesses.

This multistage process starts with contrast enhancement using a Contrast Limited Adaptive Histogram Equalization (CLAHE) algorithm on a grayscale image. This is done to amplify the contrast of faint annotations to make sure they are captured by the subsequent thresholding[9],[10]. This is done by splitting the image in a grid formation and equalizing the grayscale intensity histogram of every grid tile. To avoid over-amplifying noise, CLAHE limits the contrast by clipping the histogram at a predefined value before applying the equalization. Figure 1 shows an example.

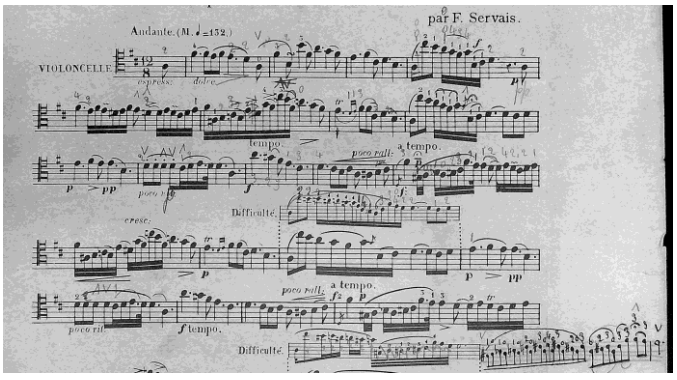


Figure 1: Music score after CLAHE

After that, both a global and local adaptive thresholding algorithm is applied to the image. Global thresholding calculates a single threshold for the grayscale intensities of pixels. If a pixel is lower than the threshold, it is considered the background, if it is above the threshold, it is considered the foreground. Local adaptive thresholding does the same,

O. Belliraj is with the Industrial Engineering Department, Ghent University (UGent), Gent, Belgium. E-mail: ossama.belliraj@UGent.be .

but instead of doing it for the entire image, it calculates a threshold for every region of the chosen kernel size. This dual approach ensures that dominant printed elements are captured by the global method, while the faint annotations are detected by the local method. This results in Figure 2, where black pixels are the foreground and white pixels are the background.

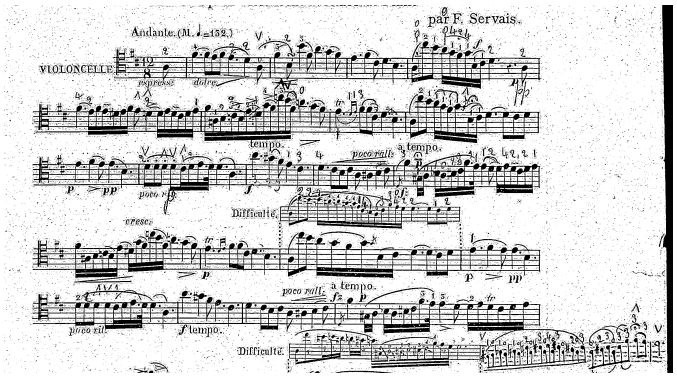


Figure 2: Music score after dual thresholding approach

Finally, both binary masks are combined into one using a logical OR operation in an inclusive strategy to preserve as much foreground information as possible. The combined mask is then refined using morphological closing to remove small noise artifacts and connect fragmented strokes. The visual result of the preprocessing pipeline is Figure 3.

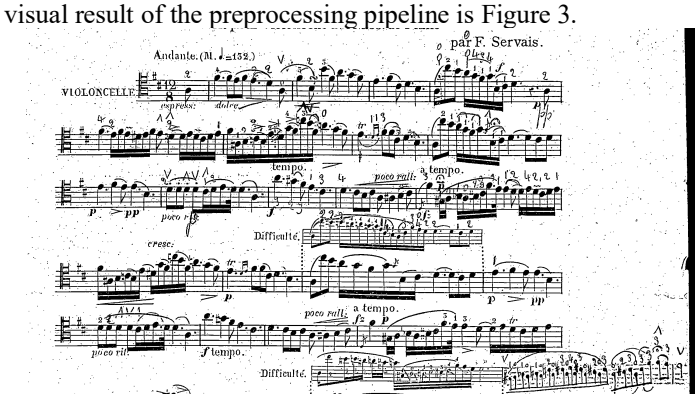


Figure 3: Music score after refining

B. *Feature extraction*

After binarization, Connected Component Analysis (CCA) is used to group adjacent foreground pixels together into individual components. For each component, a set of features is calculated. The features fall into four main categories.

Shape descriptors capture the geometric properties of the components. Printed elements tend to have a uniform shape while handwritten marks are more variable in shape. Features include Hu moments, aspect ratio, solidity, eccentricity, and compactness [12],[13].

Texture descriptors capture the surface characteristics of the components. Printed ink tends to be smooth and consistent. Handwritten ink is usually rather uneven because of the variation in pen pressure and ink flow. Features include first order intensity statistics (mean, standard deviation, entropy) and second-order statistics such as contrast, correlation, energy, and homogeneity found using Gray-level Co-occurrence matrices (GLCM) [14],[15].

Color features are features measured in both RGB and HSV spaces. The HSV space is particularly effective for distinguishing faded marks and different inks under varying illumination [16].

Edge based features such as the Sobel operator are used to describe the sharpness and continuity of components edges [11].

C. *Classification Model*

A Support Vector Machine is selected for the binary classification task as it is highly effective in high-dimensional feature spaces and is robust against overfitting [17]. By using kernel functions such as the Radial Basis Function (RBF) kernel, SVMs can learn non-linear boundaries which is needed to find subtle differences between the two classes [18].

The input of this SVM is rich feature vectors calculated using 90000 components. These components are perfectly split into 45000 components with either label 0 (printed), being components that fall mostly outside of the annotation bounding boxes, or 1 (handwritten), being components that are located fully inside the annotation bounding boxes. This has been balanced this way using undersampling as there is an overrepresentation of components with label 0. The undersampling was done per image by sorting all the components based on component area and using the number of components of the class with the least number of components in that image. The sorting was done to potentially lower the chance of using a component that is part of background noise.

IV. Results and Discussion

The proposed system was evaluated on the FAAM dataset, which contains 526 images and approximately 10000 annotations indicated with bounding boxes.

The SVM model was evaluated using three kernels with five-fold cross validation. The RBF kernel yielded the best performance with a test set accuracy of 90,52%, compared to 89.29% for the polynomial kernel and 85.97% for the linear kernel.

The RBF model demonstrates an exceptional discriminatory power with a ROC AUC of 0.96 and an Average Precision (AP) of 0.96. A detailed classification report demonstrates a balanced performance, with F1-scores of 0.9 for the printed class and 0.91 for the handwritten class.

A permutation importance analysis revealed that the standard deviation of the blue channel (b_std), entropy, equivalent_diameter, the standard deviation of the saturation (s_std), and GLCM energy (glcm_energy_asm) were the most influential features for the RBF model. This shows the SVM's ability to use complex, non-linear interactions between features.

The most significant limitation of this study is the imprecision of the ground truth. The bounding boxes often include background pixels and pixels from adjacent printed elements. This and the difficulty Connected Component Analysis has with overlapping components that are not the same, make for imprecise data given to the model. The reported accuracy should thus be taken with a grain of salt.

One way to approximate the performance of this system is a visual comparison of the ground truth of the dataset and the predictions made by the model.

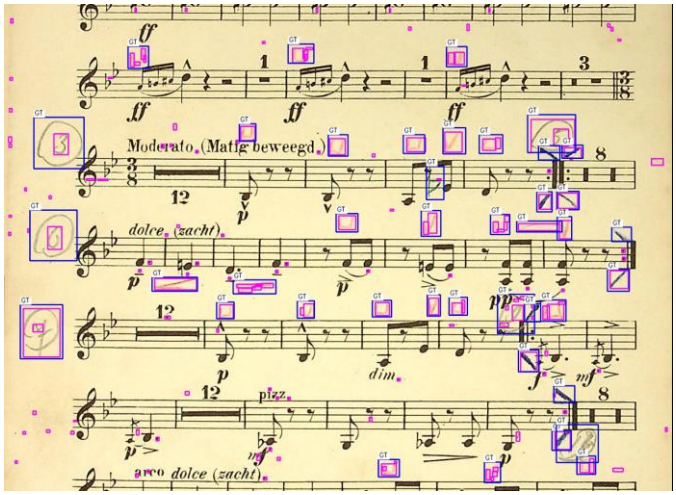


Figure 4: Comparison dataset ground truth (blue bounding boxes) and SVM model predictions (pink bounding boxes)

Multiple conclusions can be made from Figure 4. The model is good at finding components that are inside of the ground truth. This can be seen by the fact that almost every blue bounding box has a pink bounding box in them.

The three circled numbers on the left side of the image show that the model has difficulties with overlapping. The two top circles are not labeled as handwritten because of the slight overlap they have with the staff or the clef. The bottom one that is not touching the printed staff does get labeled as handwritten.

V. Conclusion

This thesis sets out to design, implement and evaluate an automated system for differentiating handwritten annotations from printed content on digitized music scores. A crucial limitation stems from the ground-truth being imprecise. This makes it hard to quantify the performance of the model created using this multistage process. This research does demonstrate that the combination of image processing techniques used offers a path toward solving the challenging problem that annotations on historical music scores pose. This system is a great baseline for future work and could achieve reliable results given a pixel-level ground truth.

References

[1] J. Calvo-Zaragoza, J. Hajič jr., and A. Pacha, "Understanding optical music recognition," ACM Computing Surveys, vol. 53, no. 4, pp. 1-35, 2020.

[2] A. Pacha and J. Calvo-Zaragoza, "End-to-end neural optical music recognition of monophonic scores," in 15th International Conference on Frontiers in Handwriting Recognition (ICFHR). IEEE, 2018, pp. 317-322.

[3] B. Gatos, K. Ntirogiannis, and I. Pratikakis, "Icdar 2009 document image binarization contest (dibco 2009)," in 10th International Conference on Document Analysis and Recognition, 2009, pp. 1375-1382.

[4] N. Otsu, "A threshold selection method from gray-level histograms," IEEE transactions on systems, man, and cybernetics, vol. 9, no. 1, pp. 62-66, 1979.

[5] F. Siddiqi and N. Vincent, "A survey of writer identification techniques," Pattern Recognition Letters, vol. 31, no. 8, pp. 619-628, 2010.

[6] N. Cornia, "Exploring plurality of interpretation through annotations in the long 19th century: musician's perspectives and the faam project," https://www.researchcatalogue.net/view/2406928/2406927, 2024, accessed on 2024-06-11.

[7] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in European conference on computer vision. Springer, 2014, pp. 740-755.

[8] N. Sharma, J. R. Saini, and P. Singh, "A comprehensive review of computational learning methods for handwritten document analysis," Archives of Computational Methods in Engineering, vol. 26, no. 4, pp. 1073-1107, 2019.

[9] J. Kimmel, B. Shbita, D. Shaked, and I. Shimshoni, "A review of historical document image enhancement," SN Applied Sciences, vol. 1, no. 8, pp. 1-22, 2019.

[10] Y. GR, "Enhancement of degraded historical document images for binarization," Journal of Electrical Systems, vol. 20, pp. 4779-4796, 08 2024.

[11] W. Alzuwawi, H. Ben Othman, and H. Hmeed, "A comprehensive review of feature extraction techniques in image processing," International Journal of Advanced Computer Science and Applications, vol. 12, no. 8, 2021.

[12] E. Kavallieratou, E. Stamatatos, N. Fakotakis, and G. Kokkinakis, "Handwritten and machine-printed text separation," Pattern Recognition, vol. 37, no. 4, pp. 859-862, 2004.

[13] B. M. Garlapati and S. R. Chalamala, "A system for handwritten and printed text classification," in 2017 1st International Conference on Next Generation Computing Technologies (NGCT), 2017, pp. 560-564.

[14] P. P. Roy, J. Lladós, and U. Pal, "Text/non-text separation from handwritten document images using lbp based features: An empirical study," in Proceedings of the 4th International Conference on Information System and Data Mining, ser. ICISDM '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 57-61.

[15] P. Shivakumara, T. Q. Phan, and C. L. Tan, "Statistical texture features based handwritten and printed text classification in south indian documents," arXiv preprint arXiv:1303.3087, 2013.

[16] H. Dasari and C. Bhagvati, "Identification of non-black inks using hsv colour space," in Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), vol. 2, 2007, pp. 486-490.

[17] C. J. Burges, "A tutorial on support vector machines for pattern recognition," Data mining and knowledge discovery, vol. 2, no. 2, pp. 121-167, 1998.

[18] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. Springer, 2009.

[19] T. Fawcett, "An introduction to roc analysis," Pattern Recognition Letters, vol. 27, no. 8, pp. 861-874, 2006.

Contents

List of Figures

List of Tables

List of Acronyms

AP Average Precision.

AUC Area Under Curve.

CCA Connected Component Analysis.

CLAHE Contrast Limited Adaptive Histogram Equalization.

FAAM Flemish Archive for Annotated Music.

GLCM Gray-Level Co-occurrence Matrix.

OCR Optical Character Recognition.

OMR Optical Music Recognition.

PR Precision-Recall.

RBF Radial Basis Function.

ROC Receiver Operating Characteristics.

SVM Support Vector Machine.

1

Introduction

The wave of large-scale digitization of cultural archives is transforming research in the humanities, offering researchers digital access to historical documents. This digital access, however, introduces a great challenge: these documents are not always simple artifacts but complex layers of information. Within the scope of musicology, historical digitized copies of printed music scores are often analyzed. These documents include an extra layer of valuable information: handwritten annotations by composers, performers and scholars. The tedious process of manually identifying and separating there layers creates a significant bottleneck for large-scale computational analysis. These annotations interesting for musicologists as they bring a certain nuance to the music piece and give an insight on the interpretation of certain parts of the piece by certain composers and performers. Standard Optical Music Recognition (OMR) tools, which are generally trained on clean, modern music scores are not equipped to navigate the complexities of historical documents. These tool have a tendency to focus on the original printed music, discarding the musicologically valuable handwritten annotations[1]. This thesis is dedicated to engineering a robust, automated system designed to accurately differentiate handwritten annotations from printed content within digitized music scores. The successful implementation of such a system would accelerate research into historical music performances.

The methodology developed in this thesis relies on RGB images from the Flemish Archive for Annotated Music (FAAM) dataset as its input. Using the provided COCO JSON files that define the approximate locations of annotations, the system initiates a multi-stage image preprocessing pipeline to segment all the foreground elements such as the handwritten annotations and everything that is printed on the page. Following the preprocessing, Connected Component Analysis is employed to isolate individual elements. These components are transformed into a feature vector that describes their color, geometry, shape, edge and textural properties. Every component is labeled as ei-

ther ”printed” or ”handwritten” based on the COCO JSON bounding boxes. This labeled feature set is the training data for a machine learning classifier tasked with learning the subtle distinctions between the classes. The entire workflow is implemented in python, primarily utilizing OpenCV and Scikit-learn libraries [2].

1.1 Objective

The core objective of this thesis is to construct and validate a system capable of classifying components from a historical music scores as either printed or handwritten by applying modern computer vision and machine learning techniques. The system must effectively handle a wide array of document degradation while accounting for the immense variability of annotations. The primary metric is maximizing the true positive rate, ensuring that a high percentage of annotations are correctly identified, while minimizing the false positive rate to prevent the misclassification of printed elements.

1.2 Related Work

A vast body of work exist in the domain of Optical Character Recognition (OCR) and Optical Music Recognition (OMR)[1]. However, this body of work is typically optimized for high quality documents and its performance diminishes drastically when confronted with the challenges posed by historical documents. For instance, many succesful OMR systems, such as that proposed by Pacha and Calvo-Zaragoza, are designed as solutions that excel at transcribing modern printed music but are not explicitly built to handle the variability of handwritten annotations overlapping with printed elements[3]. The proposed models in this body of work struggle with paper degradation, presence of overlapping, handwritten marking, which are the central focus of this research.

A sub-problem in this process is document binarization, where the goal is to seperate the foreground text and symbols from the background. While many document binarization algorithms are available, they often prove inadequate when applied to the given dataset[4]. Often used global methods like Otsu’s algorithm calculate a single intensity threshold for the entire image. This approach fails on historical documents inevitably discarding annotations in brighter areas or introduce noise in darker ones[5]. Even advanced local adaptive methods, which are evaluated in contests like DIBCO[4], can struggle, being to sensitive to noise patterns and texture of aged paper.

Other fields such as writer identification concentrate on analyzing handwriting styles.

However, as detailed by Siddiqi and Vincent, these systems presuppose that the text has already been isolated from its background and thus do not perform the separation step[6].

In summary, a gap exists for a solution specifically made to work with variable, often-degraded handwritten annotations from a printed background, a challenge this thesis aims to address.

1.3 Overview

First, an analysis of the dataset and its challenges is presented in chapter 2. In chapter 3, the complete pipeline is detailed, from preprocessing and feature extraction to the model training approach. The next chapter, chapter 4, presents and discusses the approach more concretely together with results of the pipeline, including an attempt to evaluate the binarization performance and an analysis of the trained models performance.

1.4 Use of generative AI

Generative AI has been used in the writing of this thesis. The ways it has been used are:

- Inspiration sources while writing: When not knowing how to write certain parts of the text, the idea that was hard to formulate was given to generative AI to have a more structured and clear formulation.
- Trivial code issues: While writing the code to get the results for this thesis, generative AI was used to write simple code such as code that saves images in certain directories after they were processed, or to write simple math equations need for the feature calculation.
- The lookup of certain papers that could justify an intuitive thought.
- Use of generative AI as spelling checker.
- Use of generative AI to get a better grasp of what possibilities there are in the case of my research (brainstorming). This is always paired with a nuanced look at what is generated.

2

Explorative Data Analysis

The dataset used to achieve the goal of this research is provided by the FAAM (Flemish Archive for Annotated Music). the FAAM is a database and research platfrom aiming to revive the performances of musicians from the 19th and 20th century through the study of their annotations on music scores[7].

2.1 Dataset

The dataset is a selection of music scores that have handwritten annotations on them. These music scores are scanned documents that originate from various scanners and time periods. This results in 526 image of various sizes and resolutions. These images contain about 10000 handwritten annotations. The location of these annotations are indicated by bounding boxes that have been drawn by hand and saved in a JSON file using the COCO JSON format.

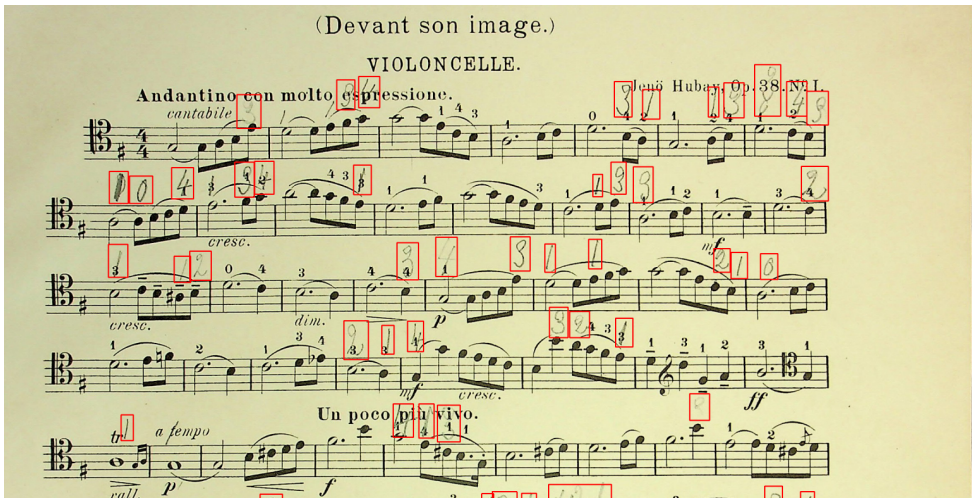


Figure 2.1: Example of music score with boundingboxes

2.1.1 COCO JSON

The COCO JSON format is a structured way of storing annotations and metadata for an image dataset, widely used in machine learning, particularly for object detection and segmentation. [8]

A COCO JSON file is structured as a dictionary that has three main sub-dictionaries:

1. Categories
2. Images
3. Annotations

Contrary to images and annotations, the categories subdictionary was not used in the FAAM Dataset. The image sub-dictionary contains information about each specific image. The data for every image are:

- Id
- filename
- height
- width

The annotation sub-dictionary holds data about every annotation and has the attributes:

- Id
- Image_id
- Category_id
- Bbox
- Segmentation

The most important of the annotation attributes are the image_id, bbox and segmentation. The image_id links the annotation to a specific image. The bbox attribute contains a list of four integers (Xmin,Ymin,width,height) indicating the location of the bounding box that highlights the annotation on the image. There is also the segmentation attribute that handles non-rectangular bounding boxes. Each polygon is a flat list [x1 ,y1 ,x2 ,y2 ,...] indicating the vertices of the polygon in order. This is important for the current research as some annotations are hard to capture in a rectangle without having other

elements of the image , such as the background and printed elements, in the bounding box.

2.1.2 Images

The dataset has 526 different RGB images of music scores. Due to the use of various scanners with various sizes and resolutions, the sizes of the images differ. Table 2.1 shows the averages and medians of the image and annotation sizes in pixels.

	Width	Height
Average image	3583.85	4593.78
Median image	2536.00	3212.00
Average bounding box	136.85	97.65
Median bounding box	60.07	67.11

Table 2.1: Image and bounding box dimensions

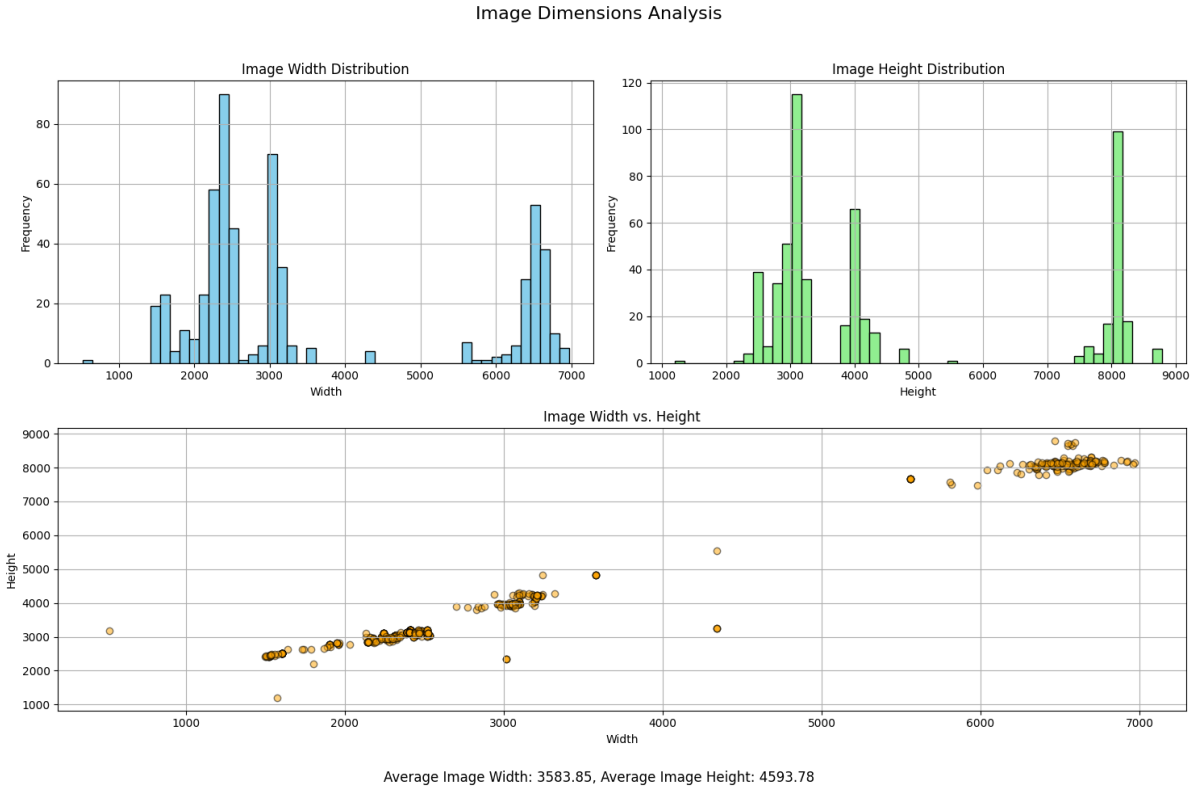


Figure 2.2: Distribution of image sizes (in pixels)

The table doesn't give the full picture of the distribution of image sizes, so their distri-

bution is shown in fig 2.2. These distributions indicate that the images can be split into three main size categories. When looking at the width, the categories would be:

- width < 2500
- width > 2500 and width < 5500
- width > 5500

Similarly, the height can be split in the categories:

- height < 3500
- height > 3500 and height < 6000
- width > 6000

The variation in image dimensions within the dataset present several challenges and considerations. These issues are:

- Computational load: Larger images demand significantly more memory and processing power. This can lead to longer processing times for tasks like feature extraction and model training.
- Information loss: Downscaling large images to a smaller uniform size can result in loss of fine detail.
- Feature scaling: The scale of features is affected by the image dimensions. This makes it harder for a model to learn generalizable patterns.

Taking these factors into account, the choice was made to keep the images with their initial dimensions. If resizing had to be done, it would require significant downscaling for a larger part of the dataset. This would mean a substantial loss of information that is necessary for the following steps. This is especially the case for the annotations that are hard to properly capture. Removing the smaller category of images to limit the downscaling of the bigger image is not an option either as this would reduce the amount images too much.

A key challenge inherent to this dataset is that these are images of historical music scores which show many signs of degradation over the years. Table 2.2 gives visual examples of the challenges that historical documents pose. It is important to note that each image can have multiple of these signs of degradations in different degrees.

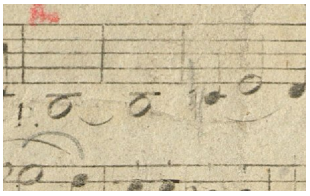
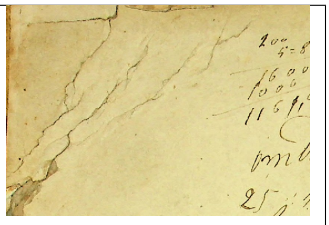
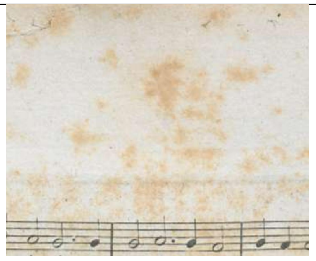
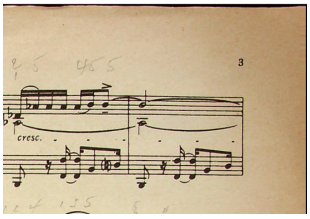
Fading ink		Physical Damage	
Noise artifacts		Uneven Illumination	

Table 2.2: Challenges of historical documents

Fading ink reduces the contrast between the foreground (musical symbols, annotations) and the background (the paper). The lack of contrast makes it harder to properly capture the annotation. The pixel intensities in fig. 2.3 show a distribution that is skewed to the right, meaning that there are more lighter pixels than darker ones. This should not be the case as the pixels used for this distribution are part of a printed staff line that is supposed to be black (low intensity value).

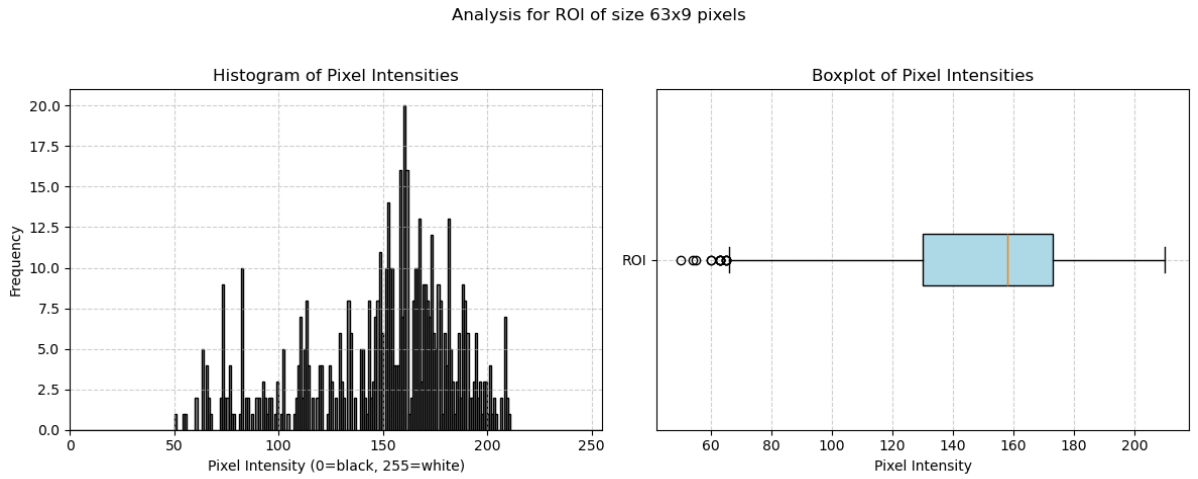


Figure 2.3: Histogram and boxplot of fading ink

Physical damage like creases and tears introduce a potential loss of information if they are in regions of interest. This physical damage can also create false positives when they are found in background regions, as they create enough contrast to be considered foreground by some algorithms.

Noise artifacts such as stains and foxing (age-related spotting) add random meaningless data tot he image. Similarly to physical damage, the noise artifacts can be misinterpreted as intentional markings.

Uneven illumination creates an inconsistent environment. Shadows can be mistaken for a dark thick line or a bright spot can be mistaken for a faded, unimportant spot leading to unreliable results across the page.

2.1.3 Annotations

Annotations are handwritten notes made on music scores by composers, musicians and musicologists for various reasons. These annotations are interesting for musicologists when studying certain music pieces as they bring nuance to what has originally been printed. Annotations come in many different sizes, shapes and colors which should make it distinguishable from the usually uniformly printed musical symbols.

Though the task of detecting these annotations may seem straightforward, there are many challenges that makes the proper segmentation and differentiation difficult. Some of these challenges can be found in table 2.3.

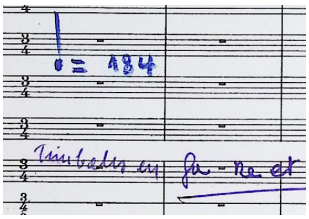
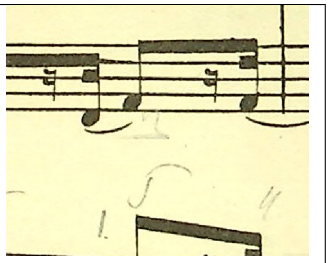
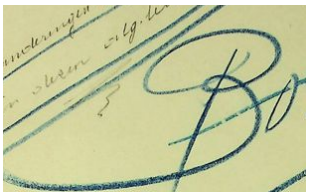
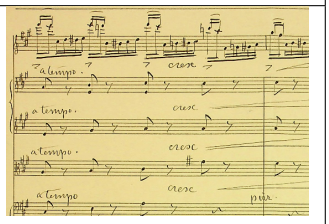
Overlap		Thin Strokes	
Handwriting variability		Hand drawn scores	

Table 2.3: Challenges of annotations

The proposed solution for this problem has made the assumption that all music scores are mechanically printed and not hand drawn.

An overlap of handwritten annotations and printed symbols is one of the biggest challenges of this dataset as it is difficult to separate the two from each other. Algorithms will

struggle to decide which pixels belong to the printed symbol and which is handwritten, something essential if the task is to classify the pixels in either one of the classes.

Annotations are often too subtle and faint, especially when the strokes are very thin. A one or two pixel wide line can easily be interpreted as image noise because of how thin it is and because of the minimal difference in grayscale intensity compared to the background. The plots in fig 2.4 show the difficulty of spotting a thin stroke on the image. The values are mostly high which would indicate the white background, yet the pixels used in this plot are from a thinly written annotation found in table 2.3

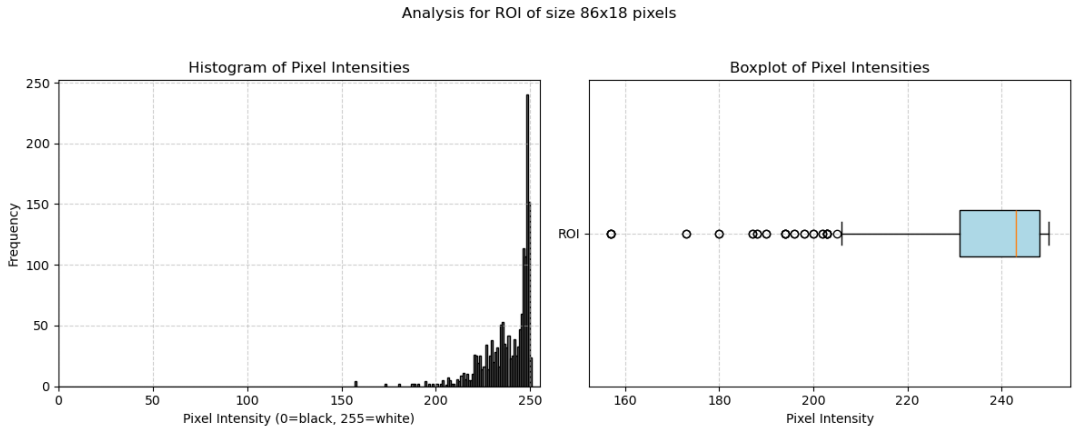
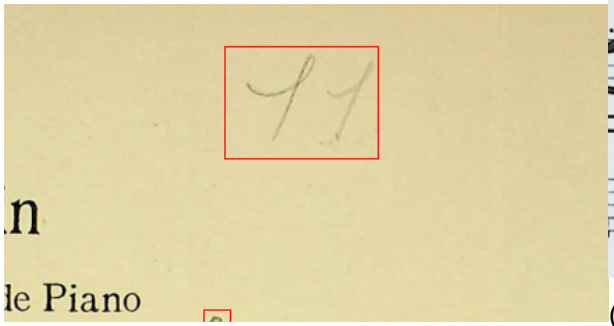


Figure 2.4: Histogram and boxplot of a thin stroke

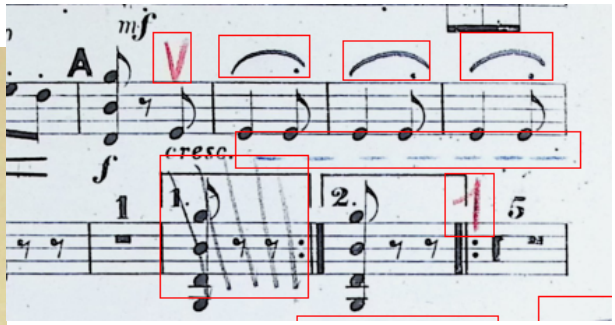
The handwriting of the annotations are very different over the whole dataset as they are written by many different people. This makes it hard to generalize them based on shape. This is even more the case with annotations as it is not only text that is written but also musical symbols, lines and random scribbles.

Hand-drawn scores make it difficult to differentiate between annotations and musical symbols as they have both been handwritten. The entire strategy of this research relies on the separation of uniform printed elements from variable handwritten elements in the image. This is why the proposed solution makes the assumption that all music scores are mechanically printed and not hand-drawn.

A signification limitation of the dataset is that the bounding boxes lack pixel-level precision. They simply define a rectangle or a polygon that contains an annotation, which means that they always include background pixels such as in fig. 2.5a, but they can also contain printed elements such as in fig 2.5b.



(a) Bounding box with background



(b) Bounding box with printed elements and background

Figure 2.5: Approximate approach of bounding boxes

3

Process

The objective of this study is to develop a methodology to differentiate pixels corresponding to handwritten annotations from those belonging to printed text in musical scores. The proposed approach consists of three main steps:

1. Preprocessing
2. Feature extraction
3. Model training

The final result of this process is a machine learning model capable of predicting whether specific pixels in an image belong to handwritten annotations, printed text, or the background. A key application of this model is in the development of an automated pipeline capable of processing images without prior annotation (i.e., without bounding boxes indicating the location of handwritten elements). The input images undergo initial preprocessing steps to facilitate the differentiation process.

3.1 Preprocessing

Preprocessing plays a crucial role in this pipeline by refining photocopies of annotated music scores to eliminate unnecessary elements, ensuring that only relevant pixels remain. Ideally, the remaining pixels are categorized into two distinct classes: those associated with printed musical symbols and those corresponding to handwritten annotations.

In essence, preprocessing aims to remove background noise from the image. A widely adopted technique in computer vision for achieving this is binarization.

3.1.1 Binarization

Binarization is the process of converting a grayscale or color image in to binary image (typically black and white)[9]. For historical music sheets, which is the focus of this research, binarization is a crucial preprocessing step because it simplifies the complex visual information, reducing the image to preferably only the necessary parts of the image, the foreground.

The primary objective of binarization in this context is to achieve an accurate segmentation of everything printed (Notes,rests,clefs,staff lines,etc) and handwritten (annotations). The difficulty in achieving this objective stems from the fact that these music sheet are old, which means they are often degraded and thus have a noisy background. Scanned documents are often unevenly lit, which leads to more difficulties. Suboptimal segmentation from foreground and background can lead to:

- a loss of subtle details
- the introduction of artifacts
- merging of certain elements, like an annotation that is written over a printed line, makes being able to differentiate between printed and handwritten symbols harder as they are merged as one

Challenges in Binarizing Historical Music Scores

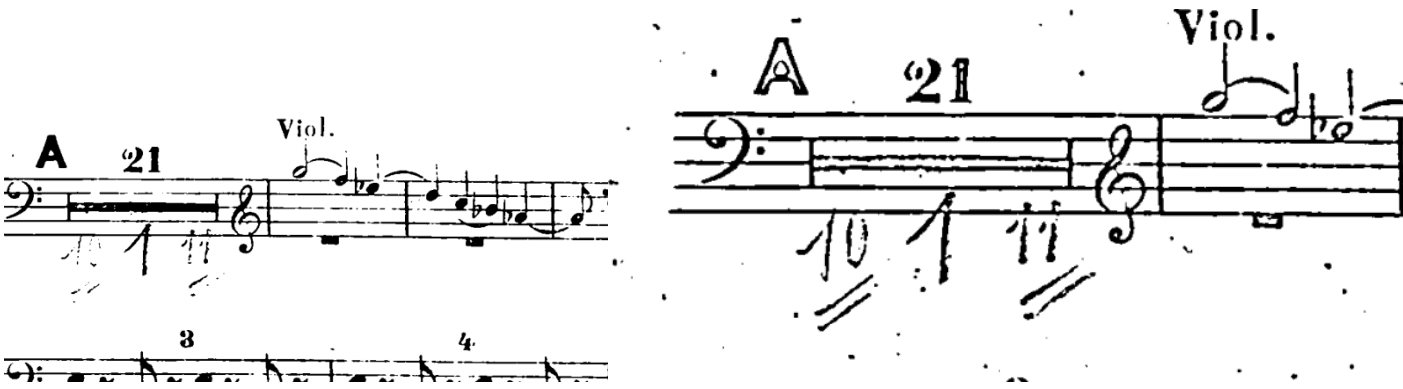
Historical music scores present a multitude of challenges for proper binarization. These challenges are very variable as in they can exist in various degrees depending on the page. The challenges are among other things uneven illumination, contrast variation, ink bleed-through, fading, smudging, background noise (foxing, stains), physical damage and so on. Each of these factors can obscure important detail or even introduce artifacts which makes the final classification task more difficult.

There are two main ways to do binarization, global thresholding or adaptive thresholding. In global thresholding, a global threshold is calculated based on the intra class variance of the grayscale of an image. Everything that is above the threshold is considered foreground and everything below it is considered background. The result of this is a binary image with white pixels (value = 255) as the foreground and the black pixels (value = 0) as the background.

The second way is called adaptive or local thresholding. Instead of calculating one

global threshold for the entire image, thresholds are calculated on local patches with a predefined kernel size. a kernelsize x kernelsize moves over the screen and calculates a threshold for only within the square.

Trying both these ways in isolation gave far from satisfactory result. When using global thresholding, some annotations have a tendency to disappear because they are more faint compared to the printed symbols, as can be seen on fig 3.1a. This results in a threshold that is too high to capture all the annotations properly. Adaptive thresholding on the other hand can create noise because of the unevenness of the background. An example of this is demonstrated on the bottom right side of fig 3.1b. When calculating a local area that only has background pixels, it will never decide to see it as only background.



(a) Otsu's unable to capture faint annotation (b) Adaptive gaussian thresholding creating more noise

Figure 3.1: Detailed difference between local adaptive and global thresholding

Due to this, a multi-stage binarization process is proposed to deal with these issues.

Overview of the proposed Multi-stage binarization process

The proposed multi-stage binarization process aims to robustly segment foreground information with an emphasis on the annotations which are inherently harder to separate from the uneven background. The process consists of 6 main steps:

- Initial conversion of the input image to a grayscale representation.
- Contrast enhancement using the Contrast Limitied Adaptive Histogram Equalization (CLAHE) algorithm.
- Application of dual thresholding strategy, employing Otsu's global thresholding method and Gaussian adaptive local thresholding.

- Combination of the binary masks resulting from the dual thresholding step via a logical OR operation.
- Refinement of the combined binary mask using morphological closing.
- A final mask inversion step to ensure standardization of output.

This multi-stage process is based on the principle that by leveraging the distinct strengths of different techniques, it is possible to systematically address the diverse and often occuring degradations found in the historical music documents.

Contrast enhancement is employed because historical documents frequently exhibit poor and uneven contrast due to the degradation caused by aging and deterioration of materials (ink and paper). Such conditions can make it harder to separate foreground elements from the background. This is even more prevalent in the annotations as handwritten elements vary greatly in how faint they are written. Some things written with a pencil can be very faint to begin with and become even more faint because of the degradation. That is why contrast enhancement is necessary.

Contrast Limited Adaptive Histogram Equalization (CLAHE) is an advanced image enhancement technique designed to address these issues by improving local contrast and enhancing visibility of details [10]. The application of CLAHE serves as an important pre-conditioner for the subsequent thresholding as thresholding techniques work optimally with very even images where the difference between foreground pixels and background pixels is very clear. This is absolutely not the case for historical documents.

CLAHE is an adaptive histogram equalization method that operates on small, typically non-overlapping regions of the image. For each of these regions it performs the following steps [11]:

- Local Histogram Computation: It calculates the histogram of pixel intensities exclusively within the boundaries of the current region
- Contrast Limiting (clipping): Before equalization, the histogram is clipped at a user-define maximum value. The contrast limit is a crucial feature that prevents over-amplification of contrast. Pixels that are over this limit are redistributed among other bins. This redistribution tries to preserve the overall brightness of the region while controlling noise. For this research, a clip limit of 3 and a grid size of 8x8 are used. This combination provided a balance between the enhancement of faint annotations and the limiting of background noise.

- Local Histogram Equalization: The clipped histogram for the region is then equalized. This process achieves a more uniform distribution of pixel intensities, effectively stretching the local dynamic range and enhancing contrast.
- Interpolation: To avoid blocky artifacts at the boundaries, bilinear interpolation between regions boundaries is applied to get a smooth transition.

The next step in the preprocessing pipeline is thresholding. Two main methods of thresholding exist.

Otsu's method is a widely recognized and extensively utilized global thresholding technique in image processing[5]. Its primary function is to automatically determine an optimal threshold value to segment the foreground and background of a grayscale image. The fundamental principle of this method is to identify a threshold value that minimizes the weighted sum of intra-class variance, which can also be seen as the maximization of inter class variance.

Adaptive thresholding automatically calculates an optimal threshold to separate the foreground from the background. The difference here is that adaptive thresholding doesn't calculate one threshold for the entire image, but calculates the threshold for every pixel based on a local region of surrounding pixels. Gaussian adaptive thresholding is a specific type of local adaptive thresholding. The threshold T(x,y) for a pixel at coordinates (x,y) is determined by calculating a weighted average of the intensities in its local neighborhood and then substracting a constant C from this average. The weighting is performed using a Gaussian kernel, which assigns higher weights to pixels closer to the center of the neighborhood.

Given the variation in image sizes across the dataset, a dynamic block size and constant C is needed depending on the image size. The The block size and C are chosen based on visual checks of different options:

- Width < 2500 : block size = 21 and C = 11
- Width > 2500 : block size = 41 and C = 21

The decision to combine the binary mask derived from global (Otsu's) and local adaptive (Gaussian) thresolding using a logical OR operation stems from the primary objective of ensuring a adequate capture of all relevant foreground information, with an emphasis on the faint and often hard to capture annotations present in the historical music scores.

Global thresolding methods are effective in clearly differentiating dominant foreground

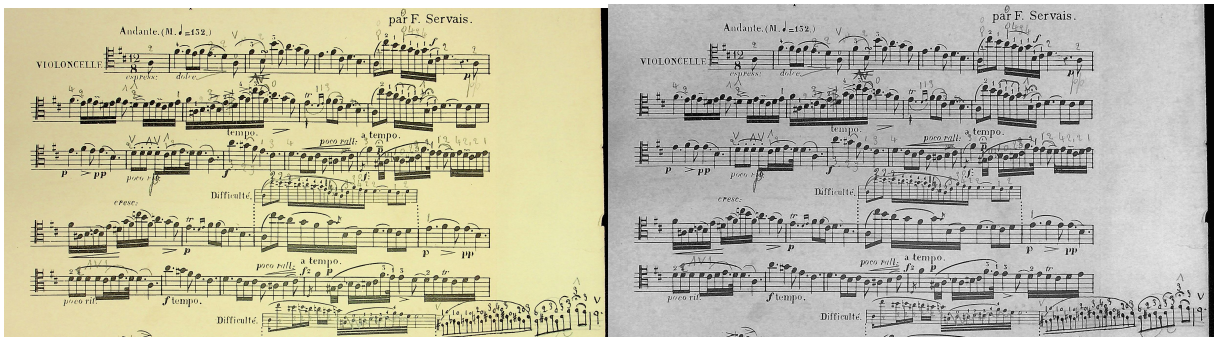
element and the general background using one threshold for the entire image. This makes it fail when faced with local variations in illumination, contrast and background textures, which are very present in this dataset.

Local adaptive methods excel precisely in the scenarios where global thresholding falters. By calculating threshold based on local neighborhoods, they can adapt to varying conditions across the image which is necessary for the faint annotations that aren't clear enough to be picked up by a global thresholding method. The local approach has it's drawbacks, as they usually introduce noise in relatively uniform local regions.

Using a logical OR operation to merge the outputs of these two methods addresses the challenge of preserving the faint details. This inclusive approach ensures that:

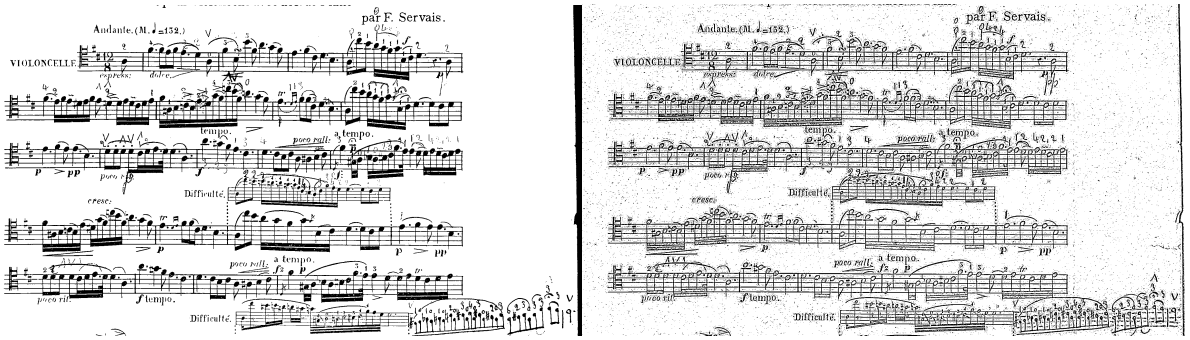
1. Strong, clear symbols robustly identified by the global method are preserved
2. Faint annotations or subtle details successfully captured by the adaptive method when it is missed by the global method

While this strategy incorporates the noise potentially being introduced by the adaptive thresholding in areas where it forces some pixels to become foreground pixels, this is a necessary trade off. To try and solve the possible noise issue, a refinement using morphological closing operation is used to remove small, isolated foreground noise pixels.



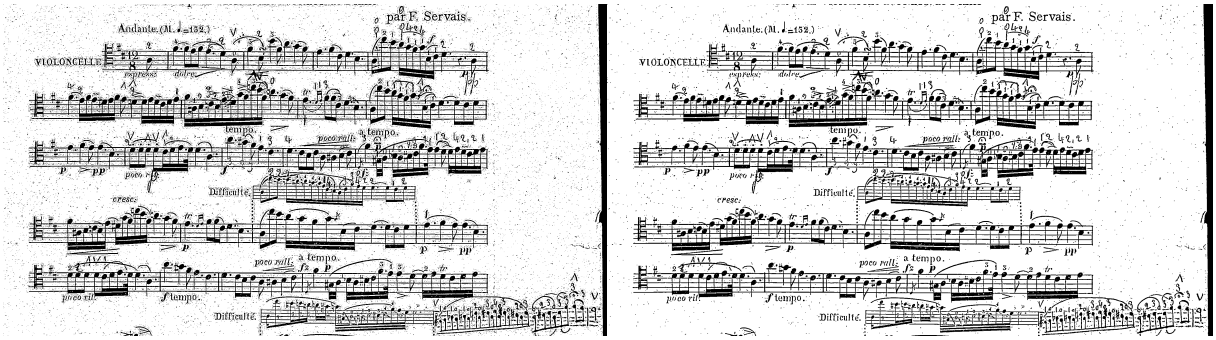
(a) Original image (b) grayscale with CLAHE applied

Figure 3.2: Step one preprocessing



(a) Otsu's thresholding applied (b) adaptive gaussian thresholding applied

Figure 3.3: Step two and three preprocessing



(a) Combined thresholding applied (b) Morphologic refining applied applied

Figure 3.4: Step four and five preprocessing

3.2 Feature extraction

Before being able to calculate features that would help differentiate printed from handwritten elements on an image, regions/pixels of interest need to be found. The first step to achieve this is done in the preprocessing, where binary masks are created for the image to indicate the foreground of the image. The foreground pixels that are in the bounding boxes are considered part of an annotation. In order to not have to work on pixel level, connected component analysis and contour detection is used group pixels of a same annotation together.

3.2.1 Connected component analysis (CCA)

Connected component analysis is a fundamental technique in image analysis. It operates on binary images, grouping pixels to each other in distinct components. Each component is then seen as an individual blob/entity.

Two connectivity types are possible:

- 4-connectivity: Pixels are considered connected if they are connected horizontally and vertically.
- 8-connectivity: Pixels are considered connected if their edges or corners touch. This means that this connectivity type adds horizontally, vertically and diagonally pixels to the component.

8-connectivity is necessary in the case of handwritten annotations because of the non-uniform shape of handwriting.

Connected components also has several limitations that are very relevant in the context of annotation detection:

- Fragmentation: Thin, faded or broken strokes often result in a suboptimal binarization, which means that some annotations will be broken up in multiple, smaller components. This can hinder the proper extraction of meaningful shape or structural features.
- Touching or Overlapping components: When handwritten annotations touch or overlap with printed components, they will be considered as one big component.
- Noise Sensitivity: Small noise components that survive the preprocessing stage

will be identified as valid connected components. This necessitates another filtering step.

3.2.2 Feature engineering for characterizing handwritten annotations

Once the foreground has been separated from the background, the next critical phase is feature engineering. This involves defining and extracting a set of features from each segmented region. The goal is to create a feature vector that represent the annotation in a way that allows the classifier to distinguish between printed and handwritten elements on a page.

Handwritten annotations on historical music scores are inherently diverse. They can range from textual comments and numbers to symbols like lines, notes and so on. They also vary when it comes to color, intensity and thickness. These annotations are on a complex background which means that having multiple kinds of features is very important for accurate differentiation.

Visual information has many different categories to draw upon to create features. The features proposed for this case can be split in four main categories:

- Shape descriptors
- Texture descriptors
- Color features
- Edge descriptors

Shape descriptors

Shape descriptors define the geometric properties of an object. For handwritten annotations, which can be very variable, shape features are necessary

Hu moments are a set of seven image moments that are invariant to translation, scale and rotation [12]. The seventh moment is also invariant to reflection. The moments are calculated from an images's central moments.

The seven moments are a combinations of centralized of centralized moments that have these invariance properties. They can be conceptually understood as follows:

- First Moment: It provides a measure of the overall spread or size of the object. Larger values of this moment correspond to shapes that are more spread out from their center
- Second Moment: The Second hu moment captures a shape's elongation. For a perfectly circular shape, this moment would be zero. Elongated shapes yield larger values
- Third Moment: This moment is sensitive to the skewness or asymmetry of the shape. A non-zero third moment can signify a lack of symmetry in the shape
- Fourth Moment: This moment describes the flatness of the shape, also known as Kurtosis
- Fifth Moment: This moment is useful for capturing more subtle details and variation in the shape's structure.
- Sixth Moment: This moment is sensitive to the relationship between the object's overall spread and asymmetries
- Seventh Moment: This moment is unique as it is invariant to reflection. This makes it useful for distinguishing, for example, the letters 'b' and 'd'

The significant variability in size, position and orientation of handwritten annotations make invariance properties very important. Hu moments can capture the fundamental shape of annotations regardless of variations, which is crucial for generalizing across different handwriting styles.

Other used shape features such as geometric features are:

- Aspect Ratio: Defined as the ratio of the width to the height of an object's bounding box
- Solidity : This is the ratio of the contour area to the area of its convex hull
- Eccentricity: This measures how much a shape deviates from being a perfect circle. A circle has an eccentricity of 0 and a straight line an eccentricity of 1
- Compactness: This feature quantifies how compact a shape is. The commonly used formula is the $Perimeter^2$ divided by the area.
- Equivalent Diameter: This is the diameter of a circle that has the same area as the object

- Orientation: This refers to the angle of major axis of an object relative to a reference axis (usually the horizontal axis)
- Perimeter: The total length of the boundary of the object. It's the distance around the outside of the shape

The use of shape descriptors is a powerful method for classification that is independent from other properties such as color and texture. Researchers have long established that features are based on the overall shape of components are effective [13]. As written by Garlapati & Chalamala, the ”character shape is unique in each font type”, whereas handwritten text shape ”depends on each individual person” [14]. This can be translated to this research because printed symbols are uniform in shape, compared to handwritten annotations that are very variable. Hu moments are important as the can identify the intrinsic shape of a component regardless of its size, position, or orientation, which can vary greatly in the case of handwritten annotations.

Texture descriptors

Texture analysis provides methods to quantify the perceived qualities of a surface, such as roughness, smoothness, and regularity, based on spatial variation of pixel intensities. In this context, handwritten strokes have different ”texture” compared to a more uniform texture of printed elements. The difference comes from factors like pen pressure variations, ink flow inconsistencies and the interaction of the pen with the paper fibers.

These features are calculated from the grayscale intensity histogram of a region. Here is a list of the used statistical features:

- Mean intensity: The average gray-level intensity of all pixels in the region. It represents the overall brightness of the area.
- Standard deviation: This measures the dispersion of pixel intensity values around the mean
- Entropy: The statistical measure of randomness in the objects intensity distribution
- Smoothness: This feature indicates the relative smoothness of the intensity in a region relative to the maximum possible intensity variation
- Uniformity/Energy: This measures the uniformity of the intensity distribution. It is

equal to the sum of the squared probabilities of each intensity level

GLCMs are powerful tools for texture analysis because they capture second-order statistics, meaning they consider the spatial relationship between pairs of pixels with specific intensity values [12].

GLCM-based features capture more complex textural patterns than first-order statistics. This makes the potentially more effective at distinguishing the subtle textural differences between pencil and print strokes.

- Contrast: Measures the local variations in the GLCM. It is high when there are large differences in intensity values between neighboring pixels.
- Correlation: Measures the linear dependency of gray levels of neighboring pixels. A high correlation value suggest a linear relationship between the intensity values of pixel pairs, indicating a more predictable ordered texture.
- Energy (Angular Second Moment): This is the measure of the local uniformity of the texture. It is calculated as the sum of the squared elements in the GLCM. A high value means more periodic and ordered texture.
- Homogeneity: Measures the closeness of the distribution of the elements in the GLCM to the GLCM diagonal

The parameters for the GLCM are pixel distance, orientation and the number of gray levels. For this thesis, the choice was made to have a pixel distance of one and this one pixel goes in the direction of 0°, 45°, 90°, and 135°. The gray levels are from 0 to 255. The angles are chosen to effectively capture fine-grained, diverse texture such as handwritten annotations. The pixel distance of one ensures that the GLCM captures local pixel relationships, which are relevant for character-level textures.

The analysis of texture is essential for differentiating the smooth, uniform texture of machine-printed ink from the variable, often uneven texture of handwritten strokes. Research looking to differentiate handwritten text from printed text effectively uses texture features due to the distinct patterns found in printed and handwritten components [15, 16]. For example, a printed staff will display high homogeneity and low contrast in its Co-occurrence Matrix (GLCM), showing a regular and predictable surface. However, a pencil or pen annotation will show greater contrast and lower energy as the writing pressure is more varied.

Color space feature (RGB,HSV)

Color information can be a powerful discriminant, and is also the first aspect people see as a good differentiating factor. Handwritten annotations have different colors depending on the pencil or pen used while the printed elements are usually in black.

The RGB color space is the standard color space for images, directly representing the intensity of red, green and blue light. The features are statistical measures like mean and variance that can be computed for each RGB channel of the annotation pixels.

A limitation of this method is that RGB values are highly correlated and are sensitive to changes in illumination. This means that the same ink might appear different RGB values under different lighting, making robust feature extraction difficult.

The HSV color space represents color information from intensity or brightness

- Hue (H): Represents the pure color. It is often stable under varying illumination.
- Saturation (S): Indicates the intensity or purity of the color (how dull or vivid it is)
- Value(V): Represents the brightness or darkness of the color

The HSV color space is often preferred for color detection and analysis due to its robustness to lighting variation. It is also particularly useful for detecting annotations that are more faded.

The use of color features, especially from the HSV space, extends beyond merely differentiating colored inks [17]. It can be crucial for distinguishing faded annotations from the document background, differentiating pencil marks from ink, and separating annotations from similarly colored degradation phenomena like foxing. This is achieved by leveraging subtle but quantifiable differences in hue, saturation, and value distributions. For example, faded ink might primarily exhibit a loss in Value (brightness) or Saturation, while its underlying Hue might remain relatively stable. Pencil marks, being graphite-based, are typically grayscale and would thus show very low saturation across all hues, contrasting with potentially more saturated ink marks. Foxing stains often have characteristic yellowish or brownish hues 1 , which can be identified by specific ranges in the HSV space. By analyzing the full distributions or statistical moments of H, S, and V components 2 , it's possible to derive signatures for different types of marks and degradation that might not be apparent in simple grayscale intensity or even RGB representations.

Edge-Based Features

Edges represent locations of significant intensity change in an image, which usually describe the boundaries of objects. These edges have characteristics such as sharpness, continuity, and density that may be able to differentiate handwritten strokes from printed elements.

The Sobel operator is a gradient-based edge detection technique that uses two 3x3 convolution kernels [12]: one detects horizontal edges and the other vertical edges. It approximates the image gradient at each pixel to calculate the magnitude of the gradient and the direction of the gradient.

Table 3.1 shows a summary of all the features.

3 Process

Table 3.1: Summary of Extracted Features

Feature Name	Category	Description
Shape Descriptors		
Hu Moments	Shape	Seven moments invariant to translation, scale, and rotation.
Aspect Ratio	Shape	Ratio of the bounding box width to its height.
Solidity	Shape	Ratio of the contour area to its convex hull area.
Eccentricity	Shape	Measures how much a shape deviates from a perfect circle.
Compactness	Shape	Quantifies shape compactness, often as Perimeter^2/Area.
Equivalent Diameter	Shape	The diameter of a circle that has the same area as the object.
Orientation	Shape	The angle of the major axis of an object.
Perimeter	Shape	The total length of the boundary of the object.
Texture Descriptors		
Mean Intensity	Texture	The average gray-level intensity of all pixels in the region.
Standard Deviation	Texture	Measures the dispersion of pixel intensity values.
Entropy	Texture	A statistical measure of randomness in the intensity distribution.
Smoothness	Texture	Indicates the relative smoothness of the intensity in a region.
Uniformity/Energy	Texture	Measures the uniformity of the intensity distribution.
Contrast	Texture	Measures local variations in the Gray-Level Co-occurrence Matrix.
Correlation	Texture	Measures the linear dependency of gray levels of neighboring pixels.
Energy (ASM)	Texture	A measure of the local uniformity of the texture.
Homogeneity	Texture	Measures closeness of the GLCM's distribution to its diagonal.
Color Space Features		
RGB Statistics	Color	Mean and variance computed for each of the R, G, and B channels.
HSV Statistics	Color	Features from Hue, Saturation, and Value; robust to lighting.
Edge-Based Features		
Sobel Operator	Edge	A gradient-based method to find edge magnitude and direction.

3.3 Model training

Following preprocessing and feature calculation, the final step in developing a method to differentiate handwritten annotations from printed symbols is to train a model. This involves training a machine learning classifier with the calculated feature vector. The efficacy of the model is based on the capacity of the model to learn complex and often subtle patterns embedded in the features.

A support vector machine (SVM) has been selected as the classifier for this binary classification. The subsequent section will provide a justification for the choice.

3.3.1 Classifier Selection: The Support Vector Machine

The differentiation between handwritten and printed text relies on a rich set of features extracted from the image. These features encompass different categories, which results in a high-dimensional feature space. SVMs are known for their effectiveness in such high-dimensional settings [18].

The distinction between handwritten and printed text is subtle and complex, which makes having a model that can make complex, nonlinear decision boundaries necessary. SVMs, especially when augmented with kernel functions, excel at this[19]. Another significant advantage, particularly in high-dimensional spaces, is SVM's robustness against overfitting, which is largely attributed to its principle of maximizing the margin between classes.

3.3.2 Core SVM Principle

The fundamental principal principle of an SVM is to calculate an optimal hyperplane that best separates the data belonging to different classes in the feature space. For binary classification tasks such as differentiating handwritten from printed, this hyperplane acts as a decision boundary. The ”optimality” of the hyperplane is defined by the calculated ”maximum margin”. The margin is the distance between the hyperplane and the closest data points of either class. The SVM algorithm aims to find the hyperplane that maximizes this margin.

The inherent design of SVMs to maximize this margin is particularly good for the nuanced task of distinguishing handwritten from printed text. Handwritten text is variable in many ways compared to printed text, which is more uniform. The feature distribution

for these two classes may overlap somewhat, which makes separation challenging. A classifier that merely minimizes training error might learn a decision boundary that is susceptible to noise or specific training data. SVMs on the other hand try to maximize the margin, which makes a more robust separation, leading to a better performance on new, unseen data.

4

Results

In this chapter, all the concepts explained in the previous chapter are applied to the FAAM dataset. First, the preprocessing will be evaluated to see how well the background has been separated from the foreground and also how well the two classes are segmented. The second step is to evaluate the calculated features to see how effective they are in differentiating both classes. Finally the model is trained and evaluated on a part of the dataset that is not used in the training.

4.1 Evaluation Preprocessing

The goal of this step is to transform the RGB images of the dataset into purely binary images that ideally separate the image into the foreground which is composed of both printed and handwritten symbols, and the rest as background.

The challenge in quantifying the performance of the preprocessing is that there is no ground truth to base the results on. As there is no example of what a perfectly segmented/binarized image is, other methods must be used to approximate the performance of the binarization.

4.1.1 Performance Annotation Detection

One way to approach this is by using the bounding boxes given in the dataset. The bounding boxes indicate the location of handwritten annotations, which have to be seen as foreground pixels after the binarization process. This means that checking if there are foreground pixels in the bounding boxes can indicate that the binarization process can capture the handwritten annotations, which are sometimes hard to capture due to them being faint.

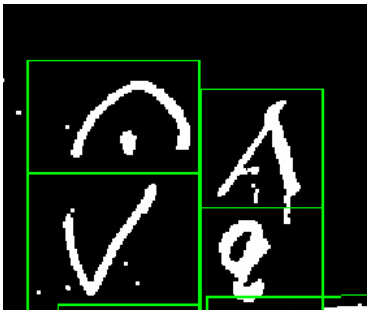


Figure 4.1: Bounding box indicating annotations

There are a couple of metrics that can be calculated to give more quantitative insights on the performance of binarization for annotations.

Quantifying annotation recall within bounding boxes

Instead of a binary check to see if there are foreground pixels in a bounding box, we can calculate the proportion of the bounding box that is correctly identified as foreground. This is done with the formula 4.1:

$$\text{Recall}_{\text{bbox}} = \frac{\text{Number of foreground pixels from binarization within the bbox}}{\text{Total area of the bbox (in pixel)}} \tag{4.1}$$

This metric can give you an idea of if the annotation has completely been captured by the binarization process. This metric does work properly only when it can be assumed that the bounding box perfectly matches with the contour of the annotation, which is usually not the case in this dataset as seen in figure 4.1.

Assessing Fragmentation of captured Annotations

Another way to assess the quality of the captured annotation is to see if the annotation has been captured as one whole blob of pixels, without disconnection. Using Connected Component analysis, two metrics can give a better view of the quality of the binarization of annotations. These two metrics are:

1. Number of Connected Components: Ideally, a single continuous annotation (like a word or a symbol) should result in a low number of connected components. A high amount of the them indicates fragmentation

2. Size of the largest Connected Component: This can indicate if the main part of the annotation is captured.

4.1.2 Experimental results

In table 4.1, the detection rate shows that the harder parts to capture in binarization, being the annotations, are being detected with a rate of 99.41%. This can mean that the binarization is happening correctly, but this could also mean that the threshold that are calculated are too much, allowing more than only the foreground to be captured(noise).

Table 4.1: Summary of Binarization Performance for Handwritten Annotations

Metric	Overall / Average	Median	Std. Dev.	Min / Max
Detection Rate (%)	99.41	–	–	–
Total Annotations Evaluated	9088	–	–	–
Total Annotations Detected	9034	–	–	–
Statistics for 9034 Detected Annotations:				
$\text{Recall}_{\text{bbox}}$	0.2376	0.2186	0.1213	0.0000 / 0.8370
Number of CCs	7.5339	3.0000	23.9774	1.0000 / 1171.0000
Largest CC Area (pixels)	1630.60	414.50	8998.48	1.0000 / 486479.0000

Note: Recall, Number of CCs, and Largest CC Area statistics are calculated across all 9034 successfully detected handwritten annotations. The Detection Rate is an overall percentage based on the 9088 total annotations processed from the evaluated images.

Looking at the recall, the average being 23.76% may indicated that the performance is not great, but as said before and nicely shown in the figure 4.1, bounding boxes generally contain a great amount of background pixels. Combining a visual check of the results of the preprocessing with the distribution seen in figure 4.2, it can be concluded that the low average does not indicate a bad performance of the binarization in function of the annotations.

4.1.3 Interpretation of preprocessing results

The issue with this approach is that this approach does not tell anything about possible noise and the performance of the binarization for the printed parts of the image although an assumption can be made that it is likely that the printed parts will be part of the foreground if the binarization performs well with the annotations.

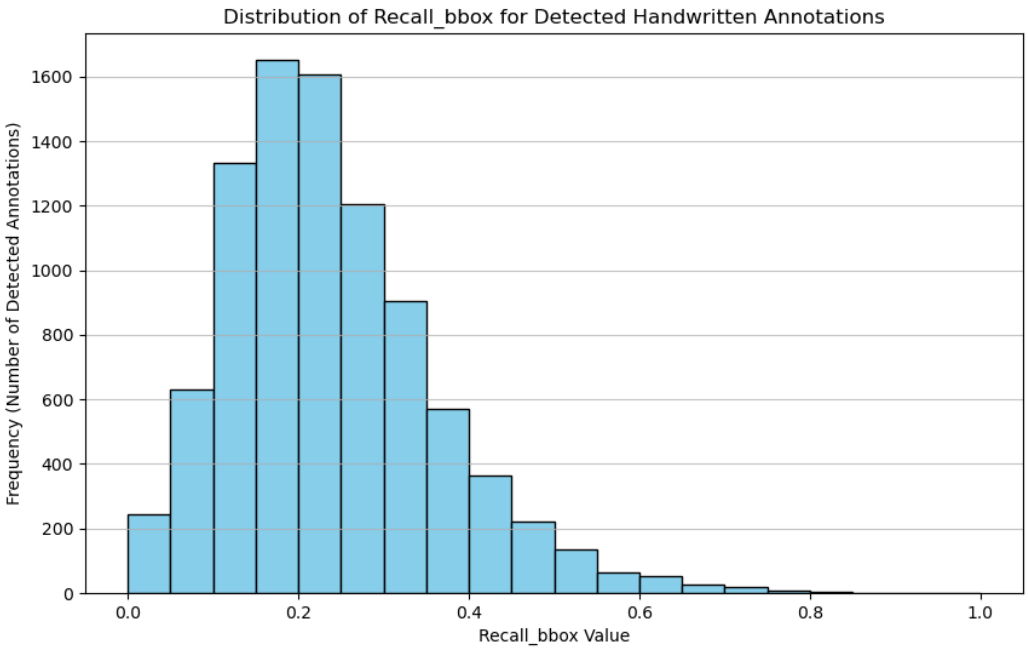


Figure 4.2: Histogram of recall_bbox

This is an issue that comes up in other areas such as OCR where the solution is to evaluate the binarization by how well it supports a downstream application. This means in this case that the binarization performs well if the model is able to properly differentiate between handwritten and printed elements on the image.

Lastly, the best way to check if the binarization is working well is by visually checking how it looks, which can take some time depending on the size of the dataset. For example Figure 4.3 shows an example that shows good binarization, seeing both handwritten and printed elements with barely any noise and fragmentation. Figure 4.4 show an example of a bad result of the binarization, as there is obviously still pretty meaningful noise to be seen

4.2 Evaluation of Features & SVM Model

Following the evaluation of the preprocessing step, the effectiveness of the the calculated features for the classification has to be measured. As indicated in the previous section, the performance of the model training is heavily linked to the performance of the preprocessing, meaning a high accuracy SVM model does not necessarily mean a

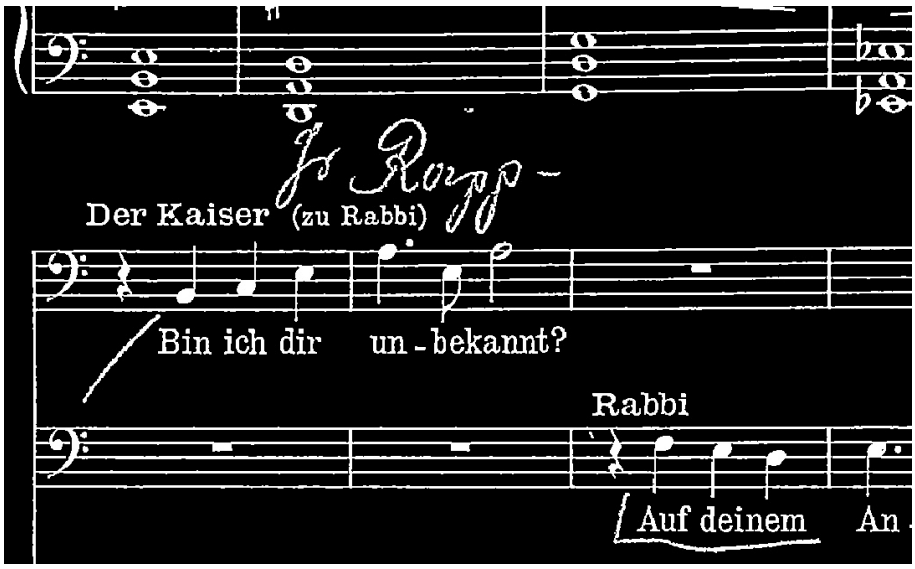


Figure 4.3: example of good binarization

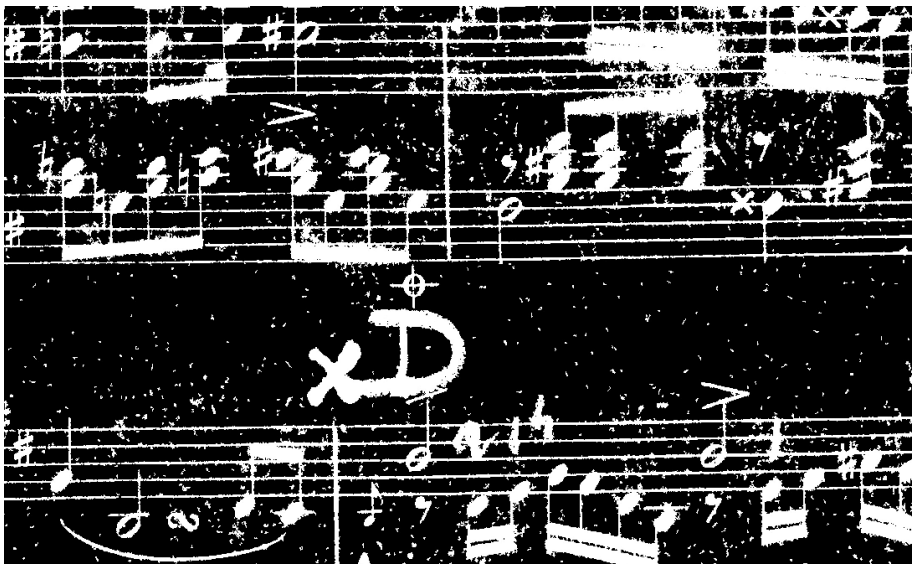


Figure 4.4: example of bad binarization

proper differentiation between printed and handwritten elements.

The calculation of the features happen on the component level instead of on pixel level. After determining the components on the image, either means and standard deviations are calculated based on every pixel of a component and shape features are computed using the contour of the component. If the component is outside of any bounding box, it is given the label 0. When a component is entirely inside of a bounding box, it is given label 1.

Due to the relatively small amount of handwritten annotations compared to either printed elements and possible noise that is most likely to arise outside the bounding boxes, using all the components would lead to a heavily unbalanced dataset. To resolve this issue and prevent noise as much as possible, undersampling is used. Depending on each amount of components per class, the minimum is used for the final dataset. To prevent noise as much as possible, the components are sorted based on component area, and only the largest components are kept when applying the undersampling.

The result of the calculating of features is a list of 90000 components with each a list of features and a label being either 0 or 1. The first step of analyzing the effectiveness of the features is to see if there distributions are properly separable.

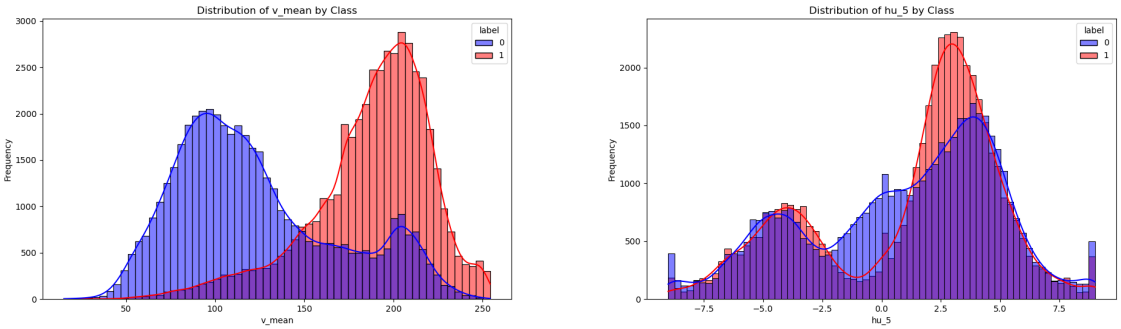
4.2.1 Quality of feature analysis

In figure 4.5, the distribution of two features are shown. The first one, the mean V value of the components shows two very separate peaks depending on the class with some overlap in the middle. This means that that the V value of a component is a good indicator for predicting if a it should have label 0 or 1. On the other hand, the distribution of the the fifth Hu moment show a great amount of overlap which would indicate a low level of linear separability.

To quantify the separability of the classes for a feature, two metrics are used.

The AUC (Area under the ROC curve) is the area found under the ROC curve [20]. The ROC curve is the plot of the true positive rate against the false positive rate. Each individual feature is treated as a classifier and the AUC measures how well that single feature can distinguish between class 0 and class 1. The AUC ranges from 0 to 1, and the closer it is to 0.5, the less it is able to distinguish between classes.

The second metric is a combination of the ANOVA F-statistic and the corresponding p-value. The F-statistic in a one-way ANOVA is a ratio of two variances, the variance



(a) Distribution of mean V value

(b) Distribution of the fifth Hu moment

Figure 4.5: Comparison of feature distributions

between classes and the variance within each class [19]. The p-value indicates the probability of observing an F-statistic as large or larger than the one calculated by the data. This means that the combination needed to have good separability would be a large F-statistics and a small p-value (<0.05).

Table 4.2: Feature Separability Analysis Results

Feature Name	**AUC**	**F-statistic**	**p-value**	**Category**
b_mean	0.856031	57404.212788	0.0000e+00	Good Separability
intensity_mean	0.855033	59787.435290	0.0000e+00	Good Separability
v_mean	0.854570	61286.024719	0.0000e+00	Good Separability
g_mean	0.851870	57809.653051	0.0000e+00	Good Separability
r_mean	0.838084	50999.686418	0.0000e+00	Good Separability
equivalent_diameter	0.831687	5601.145310	0.0000e+00	Good Separability
perimeter	0.807230	2740.437009	0.0000e+00	Good Separability
component_area_normalized	0.803090	1704.837236	0.0000e+00	Good Separability
s_std	0.790552	36235.091077	0.0000e+00	Good Separability
h_std	0.785767	19867.642445	0.0000e+00	Good Separability
v_std	0.785668	39509.554520	0.0000e+00	Good Separability
r_std	0.783808	38558.405586	0.0000e+00	Good Separability
intensity_std	0.781382	37125.220516	0.0000e+00	Good Separability
smoothness	0.781382	39.702802	2.9702e-10	Good Separability
g_std	0.780472	36590.480570	0.0000e+00	Good Separability
b_std	0.776656	33707.478325	0.0000e+00	Good Separability
color_var_gb	0.772949	5552.621950	0.0000e+00	Good Separability

Continued on next page

Table 4.2 – continued from previous page

Feature Name	AUC	F-statistic	p-value	Category
color_var_rb	0.757016	3188.979469	0.0000e+00	Good Separability
solidity	0.744743	20115.382869	0.0000e+00	Good Separability
glcm_energy_asm	0.737665	8714.140418	0.0000e+00	Good Separability
compactness	0.735907	2893.226581	0.0000e+00	Good Separability
glcm_contrast	0.735554	16126.034419	0.0000e+00	Good Separability
entropy	0.734938	15346.904892	0.0000e+00	Good Separability
color_var_rg	0.731627	8988.790228	0.0000e+00	Good Separability
s_mean	0.729805	20875.607537	0.0000e+00	Good Separability
glcm_correlation	0.703806	9792.801442	0.0000e+00	Good Separability
uniformity_energy_hist	0.631722	1114.049514	8.1260e-243	Moderate Separability
edge_density	0.602799	482.437417	1.1807e-106	Moderate Separability
glcm_homogeneity	0.574910	1538.426793	0.0000e+00	Moderate Separability
aspect_ratio	0.573934	1492.328774	4.9407e-324	Moderate Separability
hu_0	0.564458	3510.263625	0.0000e+00	Moderate Separability
hu_4	0.540997	548.914860	4.8845e-121	Low Separability
hu_3	0.528129	559.019788	3.1891e-123	Low Separability
hu_1	0.525771	773.491731	1.5637e-169	Low Separability
h_mean	0.522204	393.307841	2.3952e-87	Low Separability
hu_5	0.522190	126.316664	2.7383e-29	Low Separability
hu_2	0.521500	488.691907	5.2275e-108	Low Separability
eccentricity	0.508638	455.540072	7.8661e-101	Low Separability
orientation	0.506115	4.090438	4.3129e-02	Low Separability
hu_6	0.505083	5.871063	1.5394e-02	Low Separability

The table 4.2 shows the metrics for each of the used features and categorizes them in three different categories:

- Low separability : AUC very close to 0.5 (AUC < 0.55) and possibly a high p-value
- High separability: High AUC (or AUC > 0.7) and ideally a low p-value
- Moderate separability: AUC between high and low thresholds

The results show that 26 features show good separability, 5 show moderate separability, and the rest show low separability. What immediately becomes apparent is that the features with low separability are almost all Hu moments, while other shape features

such as equivalent diameter and perimeter have good separability. This is due to the fact that Hu moments are invariant to translation, scale and rotation which have a harder dealing with:

- Too much fragmentation due to non optimal binarization, which means that the true shape of the annotations are not captured by the features.
- The variability of the shapes of annotations makes it hard to capture a trend of similarity when it comes to shape.
- Some annotations get mixed with printed elements when turned into a component, which can give components that are seen as annotations shape features of printed elements

On the other hand, the color-based and grayscale intensity features seem to perform the best on that front, which was to be expected, as printed elements are always black, but annotations come in many different colors.

The fact that some features are not linearly separable does not necessarily mean that they cannot be used for this classification. Due to the kernel trick available for SVMs, these features can still have some impact on the prediction of the final model.

4.2.2 SVM model training results

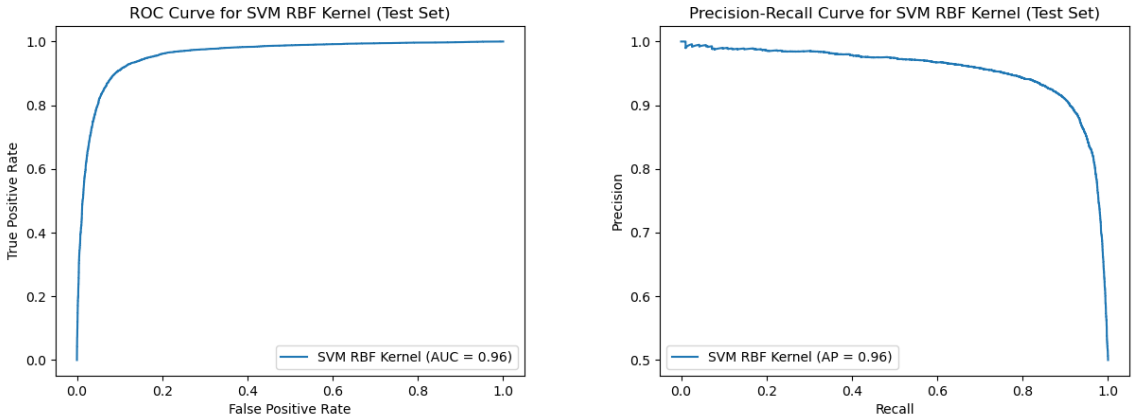
After checking the quality of the features, the features are fed to the SVM model to train it and make it capable of finding components on the image that are annotations. SVMs are popular model for classification because they are robust and have a way to work with non-linear data. The dataset has been trained on using three different kernels: Linear, polynomial and radial basis function (RBF).

The data set is divided into five configurations of 80% training data and 20% test data to perform a five-fold cross-validation. The training accuracy is also compared to the cross-validation accuracy to check for overfitting.

The results in table 4.3 show that the overall accuracies are high for each of the used kernels, with the RBF kernel having the best accuracy. The better performance of the RBF and polynomial kernels is probably due to the features in the dataset that had bad linear separability. As the RBF kernel performs the best, the rest of the results will be shown using this kernel.

Table 4.3: SVM Kernel Evaluation

Kernel Type	Cross-Validation Acc.	Training Set Acc.	Test Set Acc.	Overfitting Check
SVM RBF	0.9065 ± 0.0036	0.9106	0.9052	0.0041
SVM Poly	0.8952 ± 0.0032	0.9003	0.8929	0.0051
SVM Linear	0.8614 ± 0.0050	0.8616	0.8594	0.0002



(a) ROC curve SVM model

(b) PR Curve SVM Model

Figure 4.6: ROC and PR curve of the trained SVM RBF model (Test set)

In-depth performance Analysis of the SVM RBF Model

Following the model selection process, where the SVM with a RBF kernel was identified as the optimal performing model, a more in-depth analysis is conducted to properly evaluate the predictive capabilities on the test dataset.

Overall predictive accuracy The SVM RBF model show a strong performance on the unseen test, achieving an overall accuracy of 90.52% which mean that the model classified 9 out of 10 components correctly.

Further metrics to determine between classes was conducted using Receiver Operating Characteristic (ROC) analysis and Precision-Recall (PR) analysis. The Area Under the ROC Curve (AUC), also used to determine the linear separability of the classes per feature, results in a 0.9583, which is an excellent score. Complementing this, the Average Precision (AP) from the precision-Recall curve is 0.9550. A visual representation of the area's under the curve can be seen in figure 4.6

Class-wise Performance and Error Analysis To understand the model's performance for each class, a confusion matrix and detailed classification report can give necessary insights.

Table 4.4: Confusion Matrix for SVM RBF Kernel (Test Set)

		Predicted Class	
		0	**1**
Actual Class	**0**	8300 (TN)	971 (FP)
	1	787 (FN)	8484 (TP)

In table 4.4, we can observe that:

- The model correctly identified 8300 instances as Class 0 (True Negatives) and 8484 instances as Class 1 (True positives)
- There were 971 instances of Class 0 incorrectly predicted as Class 1 (False Positives)
- There were 787 instances of Class 1 incorrectly predicted as Class 0 (False Negatives)

The classification report provides further per-class metrics:

Table 4.5: Classification Report for SVM RBF Kernel (Test Set)

Class	**Precision**	**Recall**	**F1-score**	**Support**
0	0.91	0.90	0.90	9271
1	0.90	0.92	0.91	9271
Accuracy			0.91	18542
Macro Avg	0.91	0.91	0.91	18542
Weighted Avg	0.91	0.91	0.91	18542

The results from table 4.5 indicate a well-balanced performance across both classes:

- For class 0, a precision of 0.91 means that when the model predicts an instance as class 0, it is correct 91% of the time. The recall of 0.90 signifies that the model successfully identified 90% of all actual class 0 instances.

- For class 1, the precision is one percent lower than class 0, and the recall two percent higher. The F1-scores, which balances precision and recall, are 0.9 for class 0 and 0.91 for class 1, underscoring the consistent performance.

Feature Importance To gain insight in which features most influenced the SVM RBF model's prediction, permutation importance was calculated on the test set. Permutation importance is a technique used to understand how much each feature contributes to a model's performance. The basic idea is to measure the importance of a feature by observing how much the model's performance decreases when that feature's value is made random. If a feature is important, randomly changing it's values should decrease the model's accuracy.

The top 5 most influential features, along with their mean importance scores and standard deviations across repeats. The standard deviation indicates the stability of these importance scores.

1. b_std: 0.0278 (0.0012)
2. entropy: 0.0250 (0.0016)
3. equivalent_diameter: 0.0234 (0.0008)
4. s_std: 0.0206 (0.0011)
5. glcm_energy_asm: 0.0182 (0.0010)

The distribution of importances for all features is visualized in Figure 4.7. This plot allows for a comparison of the relative importances across the features.

It is insightful to compare these permutation importances with the earlier feature separability analysis in Table 4.2. While the separability analysis identified mostly color and grayscale intensity features as having high discriminatory power, they appear less in the top permutation importances for the SVM RBF model. Similarly, features such as solidity and entropy which scored lower than the color features when it comes to separability, are more important to this specific SVM RBF model.

The difference highlights certain aspect of feature importance:

- The linear separability measures the intrinsic ability of a single feature to distinguish classes in isolation
- Permutation importance reflects how much the trained SVM RBF model, which

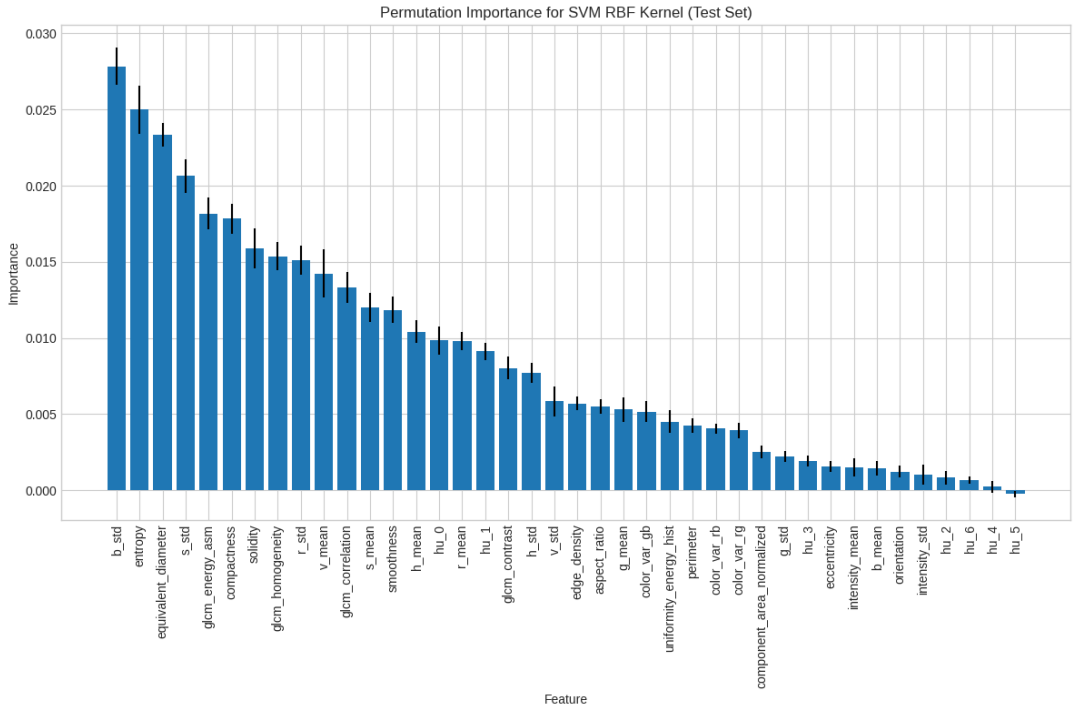


Figure 4.7: Feature importance for SVM RBF model

can handle non-linear relationships and feature interactions due to its kernel, relies on each feature

4.3 Summary and Reflection on Results

While the quantitative metrics suggest great results, it is important to interpret these results with caution. As cited multiple times, the dataset's ground truth consists of approximate bounding boxes that lack pixel level precision. This imprecision means that the training labels are ambiguous. Therefore, a visual representation is necessary to properly evaluate the results and understand the practical implications of this limitation.

To achieve this visualization, an image goes through the preprocessing, component-level segmentation, and feature extraction. The feature vector of each component is fed to the model, resulting in either label 0 or label 1. Bounding boxes are drawn around the components that were given label 1.

Figure 4.8 shows an example of the predictions the model is capable of. This visualization can give insight on how the model works and what its limitations are:

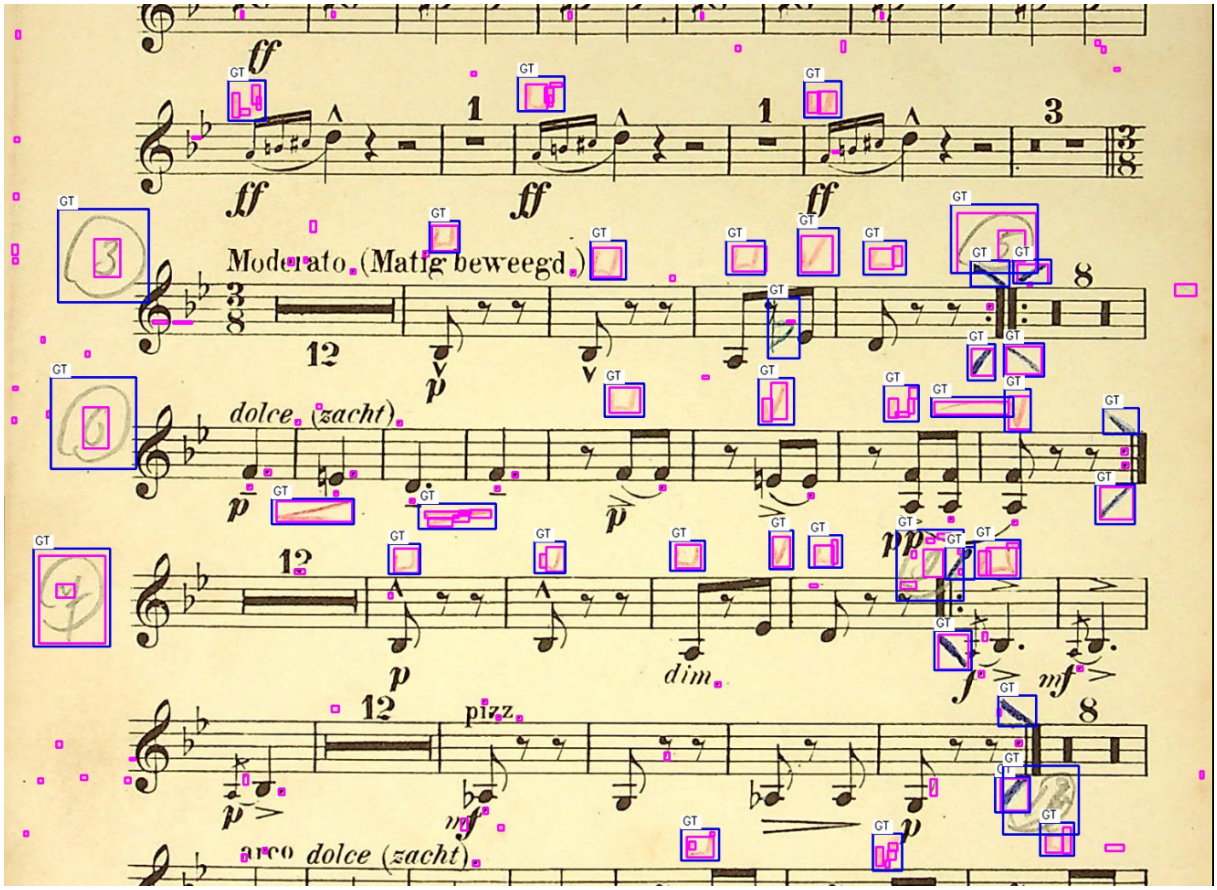


Figure 4.8: Example of model prediction (blue=ground truth,pink=prediction)

- There is significant overlap between the predictions and the ground truth, the prediction usually being smaller as they are more accurately drawn around the component
- There is still some noise visible, but these noisy components have a tendency to be very small
- Some annotations suffer from fragmentation, as can be seen by the multiple pink bounding boxes in one single blue bounding box
- As can be seen in the three big annotations on the left, which represent circled number. Only the annotation at the bottom has a predicted bounding box around the circle. This is the case because the other two circles are connected to the staff line next to them, making them one big component. This is a limitation of the Connected Component Analysis, which could be solved by a pixel-level approach given a pixel-level ground truth.

In conclusion, the visual reconstruction cannot show a perfect separation because a perfect ground truth does not exist for this dataset. Instead, it provides an insight into the model's behavior. It confirms that the model has learned between the two classes provided by the data. It also underscores the conclusion that the reliability of the result is constrained by the the imprecision of the bounding boxes. A truly accurate pixel-level evaluation require a pixel-level ground truth.

5

Conclusion

This thesis set out to design, implement, and evaluate an automated system for differentiating handwritten annotations from printed symbols on a digitized historical music score. To achieve this, a three-stage pipeline is proposed which include: a multi stage binarization process tailored for degraded documents, a comprehensive feature extraction step and a classification stage using a Support Vector Machine (SVM).

The final results show the promise of this approach. A crucial limitation stems from the ground-truth data: Bounding boxes often encapsulate not only target annotations but also neighboring printed elements. Consequently, the binarization and components analysis stages can introduce 'impure' components containing a mix of handwritten and printed pixels. While the binarization method successfully captured foreground pixels within 99.41% of the bounding boxes, this metric does not show the purity of the resulting components. This data ambiguity carries through to the classification stage. While the SVM model's performance is notable using an RBF kernel, the performance is done using the partially impure components, adding possible noise and making the decision boundary not 100% reliable.

In conclusion, this research demonstrates that the combination of image processing techniques used offers a path towards solving the challenging problem of separating handwritten and printed content in historical documents. This system provides a great baseline for future work and could achieve better and more reliable results given a pixel-level ground truth.

6

Future Work

The research presented in this thesis established a first attempt at a system for the differentiation of handwritten annotations from printed content in historical music scores. Despite the inaccuracy of the ground truth, this research successfully made a first step in solving the challenges that this topic brings. Future works can work in enhancing either the dataset, or one of the stages of this multi stage solution as the performance of each of the stages heavily relies on the other stages.

6.1 Enhancement of ground truth

A clear limitation throughout this research is the reliance on an approximate ground-truth. This inherently limits both the performance of the classification model and the capability to properly evaluate the stages of the proposed process. The most impactful future work would be the creation of a pixel-level ground truth dataset. This would involve segmenting images to label each pixel as either handwritten, printed, or background. Such a dataset would:

- Enable training and evaluation of more sophisticated, pixel-level classification models such as U-Net
- Allow a more accurate evaluation of the preprocessing stages
- Resolve ambiguities for components that contain both printed and handwritten elements, leading to a more robust classification

6.2 Advanced Segmentation and Component Analysis

The current methodology relies on Connected Component Analysis to segment the foreground pixels. While being relatively effective, CCA struggles with fragmentation and overlapping. Future works could explore more advanced segmentation techniques, given a more accurate dataset, such as:

- Deep Learning Segmentation models trained on pixel-level ground truth could separate overlapping handwritten and printed elements.
- Advanced Morphological Operations could help with a better reconstruction of fragmented strokes without merging with nearby components.

6.3 Broadening the Scope and Application

The proposed system provides a baseline that can be extended to broader contexts:

- Application to other historical documents, not only limited to music scores. The techniques used in this research could be tested on historical manuscripts, legal documents, or personal letters. This could be more successful as these types of historical document usually do not as much overlapping, something the proposed model struggles with.
- Semantic Analysis of the annotations. When a good working system is created, an attempt could be made to not only find the location of the annotation on the page, but also know why it's there and what it means.

7

Societal Reflection

The work presented in this thesis carries significant implications for sustainability, viewed through the lens of cultural preservation, digital transformation and responsible AI development. This section reflects on these aspects and placing them within a broader societal context aligned with the United Nations Sustainable Development goals.

7.1 Cultural Preservation

This project is at its core an act of cultural sustainability. Historical music scores are artifact susceptible to degradation over time. By developing a robust way to digitize not only the printed notes but also the handwritten annotations, this work contributes to the preservation of invaluable cultural heritage. These annotations give contain unique insight into historical performances that are otherwise lost to time.

This aligns with UN SDG 11: Sustainable Cities and Communities which calls for an effort to protect and safeguard the world's cultural and natural heritage. By creating tools that make this heritage more accessible and usable for musicologists, this research helps ensure that theses cultural assets can be studied and kept.

7.2 Contribution to Digital Transformation and Responsible AI

This thesis contributes to the broader digital transformation of the humanities. It demonstrates how machine learning can be applied to complicated problems that would otherwise require time-consuming manual labor. By automating a part of musicological

research, this empowers researchers to analyze historical sources at a scale that was previously not possible.

A conscious decision was made to employ a classical machine learning approach (Support Vector Machine with engineered features) rather than a deep learning model. Deep learning models are resource intensive, requiring significant computational power and energy for training, which translates to a larger environmental footprint.

In conclusion, this project represents a sustainable application of AI, one that focuses on preserving the past to enrich the future. It provides a valuable tool for the digital humanities and serves as an example of how technology can be used to aid and enhance other fields of research.

References

[1] J. Calvo-Zaragoza, J. Hajič jr., and A. Pacha, “Understanding optical music recognition,” *ACM Computing Surveys*, vol. 53, no. 4, pp. 1–35, 2020.

[2] O. Belliraj, “Thesis-2025: Source Code for Feature-Based Detection of Handwritten Annotations on Historical Sheet Music,” https://github.com/ossama-belliraj/Thesis-2025, 2025, accessed: June 12, 2025.

[3] A. Pacha and J. Calvo-Zaragoza, “End-to-end neural optical music recognition of monophonic scores,” in *15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 2018, pp. 317–322.

[4] B. Gatos, K. Ntirogiannis, and I. Pratikakis, “Icdar 2009 document image binarization contest (dibco 2009),” in *10th International Conference on Document Analysis and Recognition*, 2009, pp. 1375–1382.

[5] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[6] F. Siddiqi and N. Vincent, “A survey of writer identification techniques,” *Pattern Recognition Letters*, vol. 31, no. 8, pp. 619–628, 2010.

[7] N. Cornia, “Exploring plurality of interpretation through annotations in the long 19th century: musician’s perspectives and the faam project,” https://www.researchcatalogue.net/view/2406928/2406927, 2024, accessed on 2024-06-11.

[8] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[9] N. Sharma, J. R. Saini, and P. Singh, “A comprehensive review of computational learning methods for handwritten document analysis,” *Archives of Computational Methods in Engineering*, vol. 26, no. 4, pp. 1073–1107, 2019.

[10] J. Kimmel, B. Shbita, D. Shaked, and I. Shimshoni, “A review of historical document image enhancement,” *SN Applied Sciences*, vol. 1, no. 8, pp. 1–22, 2019.

[11] Y. G R, “Enhancement of degraded historical document images for binarization,” *Journal of Electrical Systems*, vol. 20, pp. 4779–4796, 08 2024.

[12] W. Alzuwawi, H. Ben Othman, and H. Hmeed, “A comprehensive review of fea-

ture extraction techniques in image processing," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 8, 2021.

[13] E. Kavallieratou, E. Stamatatos, N. Fakotakis, and G. Kokkinakis, "Handwritten and machine-printed text separation," *Pattern Recognition*, vol. 37, no. 4, pp. 859–862, 2004.

[14] B. M. Garlapati and S. R. Chalamala, "A system for handwritten and printed text classification," in *2017 1st International Conference on Next Generation Computing Technologies (NGCT)*, 2017, pp. 560–564.

[15] P. P. Roy, J. Lladós, and U. Pal, "Text/non-text separation from handwritten document images using lbp based features: An empirical study," in *Proceedings of the 4th International Conference on Information System and Data Mining*, ser. ICISDM '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 57–61.

[16] P. Shivakumara, T. Q. Phan, and C. L. Tan, "Statistical texture features based handwritten and printed text classification in south indian documents," *arXiv preprint arXiv:1303.3087*, 2013.

[17] H. Dasari and C. Bhagvati, "Identification of non-black inks using hsv colour space," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, vol. 2, 2007, pp. 486–490.

[18] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[19] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. Springer, 2009.

[20] T. Fawcett, "An introduction to roc analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.