



UNIVERSIDADE FEDERAL DO MARANHÃO

Curso de Ciência da Computação

Gustavo Ântonny de Sousa da Silva

**Reconhecimento Automático de Partituras
Manuscritas Utilizando SVM e ORB**

São Luís - MA

2017

Gustavo Ântonny de Sousa da Silva

Reconhecimento Automático de Partituras Manuscritas Utilizando SVM e ORB

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. João Dallyson Sousa de Almeida

São Luís - MA

2017

Sousa da Silva, Gustavo Ântonny de.

Reconhecimento automático de partituras manuscritas
utilizando SVM e ORB / Gustavo Ântonny de Sousa da Silva.
- 2017.

75 f.

Orientador(a): João Dallyson Sousa de Almeida.

Monografia (Graduação) - Curso de Ciência da
Computação, Universidade Federal do Maranhão, São Luís,
2017.

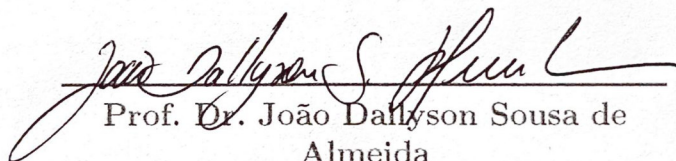
1. BoK (Bag of KeyPoints). 2. OMR. 3. ORB. 4.
Partituras Musicais Manuscritas. 5. Reconhecimento de
Imagens. I. Almeida, João Dallyson Sousa de. II. Título.

Gustavo Ântonny de Sousa da Silva

Reconhecimento Automático de Partituras Manuscritas Utilizando SVM e ORB

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

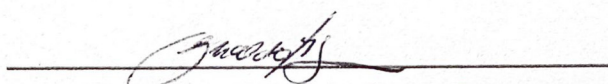
Trabalho Aprovado em: São Luís - MA, 14 de Novembro de 2017:



Prof. Dr. João Dallyson Sousa de
Almeida

Orientador

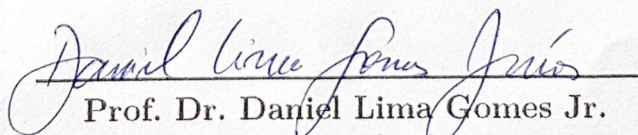
Universidade Federal do Maranhão



Prof. Dr. Geraldo Braz Junior

Examinador 1

Universidade Federal do Maranhão



Prof. Dr. Daniel Lima Gomes Jr.

Examinador 2

Instituto Federal do Maranhão

São Luís - MA

2017

Agradecimentos

Meu sincero agradecimento, em primeiro lugar a Deus, que me ensinou a acreditar nas oportunidades do impossível. À minha querida família, meus pais Pacífico e Sâmý (Minha mais profunda e genuína gratidão eterna...) e meu irmão Alexandre (Humrum, sei... Kkkkk) que sempre se doaram infinitamente por mim, para que eu pudesse entender o que realmente importa na vida, Obrigado.

Ao meu orientador, Prof. Dr. João Dallyson Sousa de Almeida, pela oportunidade concedida e seu apoio inestimável que me inspirou em todos momentos da minha graduação, sempre motivador, disponível, paciente e ensinando mesmo fora da sala de aula.

Aos meus professores, Prof. Dr. Geraldo Braz Junior, Prof. Dr. Carlos Salles, Prof. Dr. Anselmo Cardoso de Paiva que me inspiraram, aconselharam e que colaboraram com muito mais do que meu conhecimento acadêmico, os quais terão meu eterno agradecimento e admiração.

*“A música dá alma ao universo, asas à mente,
voo a imaginação, e vida à tudo!
(Platão)*

Resumo

O reconhecimento de imagens necessita de um esforço especial no sentido de trabalhar com símbolos musicais manuscritos, o processo automatizado de digitalização e reconhecimento de partituras musicais manuscritas, uma área do OMR (Optical Music Recognition), é um problema conhecido e estudado na área de processamento de imagens. Entretanto, até o momento, ainda há a necessidade de novas propostas para o problema, especialmente quando se lida com a notação manuscrita. Portanto, esta produção, tem o intuito de apresentar uma metodologia para a sistematização do processo de reconhecimento de partituras manuscritas, abordando as principais técnicas presentes nas áreas da visão computacional, análise de documentos, reconhecimento de padrões entre outras.

Neste trabalho, foi utilizado o algoritmo ORB (Oriented FAST and Rotated BRIEF), para a aquisição das características sob uma base de imagens com partituras manuscritas digitalizadas, utilizando os métodos de Brute Force Matcher, BoK (Bag of KeyPoints) e SVM (Support Vector Machine) para realizar a classificação e validação dos símbolos musicais. Obtendo resultados que atingiram 98,050% de acurácia para a classificação de 20 classes de símbolos, 100% na etapa de detecção de pautas, 98,28% no processo de remoção das pautas, 86,12% no estágio de identificação das posições das notas e 66,12% no processo de reconhecimento das barras de compasso.

Palavras-chaves: OMR; Partituras Musicais Manuscritas; Reconhecimento de Imagens; ORB; BoK (Bag of KeyPoints); SVM (Support Vector Machine)

Abstract

The image recognition requires a special effort to work with handwritten musical symbols, the automated process of scanning and recognizing handwritten musical scores, an area of the OMR (Optical Music Recognition), is a well-known and studied problem in the area of processing images. However, to date, there is still a need for new proposals for the problem, especially when dealing with handwritten notation. Therefore, this production intends to present a methodology for the systematization of the process of recognition of handwritten scores, addressing the main techniques present in the areas of computer vision, document analysis, pattern recognition among others.

In this work, the ORB (Oriented FAST and Rotated BRIEF) algorithm was used for the acquisition of the features in an image-based scanned handwritten scores, using Brute Force Matcher, BoK (Bag of KeyPoints) and SVM (Support Vector Machine) to perform the classification and validation of musical symbols. Obtaining results that reached 98.050% accuracy for the classification of 20 symbol classes, 100% in musical staff pattern detection stage, 98.28% in the process of removing the musical staff, 86.12% in the identification stage of the positions of the notes and 66.12% in the bar line recognition process.

Keywords: OMR; Handwritten Musical Scores; Image Recognition; ORB; BoK (Bag of KeyPoints); SVM (Support Vector Machine)

Lista de ilustrações

Figura 1 – Notas na Clave de Sol	22
Figura 2 – Tetragrama	23
Figura 3 – Cifras	24
Figura 4 – Um exemplo da notação digital atual, Prelúdio, Op. 28, No. 7, Frédéric Chopin.	25
Figura 5 – Versão manuscrita da Sonata para piano e violoncelo, Op. 65, Frédéric Chopin	25
Figura 6 – Arquitetura OMR	27
Figura 7 – Ilustração do funcionamento do algoritmo Flood Fill	34
Figura 8 – Todas as classes com seus respectivos símbolos utilizados nessa pesquisa.	35
Figura 9 – Detector FAST: Exemplo do ponto de interesse p, e os 16 pixels no círculo de teste.	36
Figura 10 – BRIEF: Exemplo de cinco padrões de amostragem.	37
Figura 11 – Ilustração de um hiperplano ótimo para dois conjuntos de dados	42
Figura 12 – Pipeline representando as seis etapas propostas para o reconhecimento de uma partitura manuscrita.	49
Figura 13 – Mostra o antes (a) e depois (b) da imagem da partitura escolhida, na qual foi empregada o método de binarização de Otsu, e a operação morfológica de dilatação.	51
Figura 14 – A imagem apresenta em vermelho as falhas, os gaps comumente encontrados nas linhas de pauta manuscrita.	51
Figura 15 – Apresenta a mesma imagem após ser efetuada a dilatação.	51
Figura 16 – Os segmentos de pauta detectados estão em vermelho.	52
Figura 17 – Detecção das linhas de pauta em vermelho.	53
Figura 18 – Partitura após etapa de remoção das pautas.	54
Figura 19 – Símbolos da partitura segmentados em vermelho.	55
Figura 20 – Pipeline para a detecção dos símbolos musicais manuscritos utilizando o ORB. Os símbolos foram extraídos na etapa de pré-processamento.	56
Figura 21 – Grid-Search - Parâmetros $C = 8$ e $\gamma=0.0078125$, com uma acurácia de 98.0625%.	57
Figura 22 – Exemplo da matriz de confusão para o melhor resultado encontrado.	60
Figura 23 – A figura apresenta em vermelho os símbolos da partitura classificados como semínima	62
Figura 24 – Principais componentes da estrutura primitiva de uma nota musical, reconhecidas como: hastes, cabeças e colchetes.	63

Figura 25 – Versão final das informações que foram identificadas e classificadas na partitura manuscrita submetida a metodologia apresentada.	65
Figura 26 – Partitura recuperada, em sua versão final, convertida do formato MusicXML gerado pelas metodologias abordadas nesse estudo.	67

Lista de tabelas

Tabela 1 – Algumas das principais características da notação musical.	26
Tabela 2 – A tabela apresenta os principais kernels utilizados em aplicações com máquinas de vetores de suporte.	43
Tabela 3 – Símbolos musicais utilizados nesse trabalho.	47
Tabela 3 – Símbolos musicais utilizados nesse trabalho.	48
Tabela 4 – Apresenta os dados para a etapa de estimação dos parâmetros do kernel RBF, bem como os resultados da classificação dos símbolos musicais manuscritos, para cada etapa da análise da validação cruzada.	58
Tabela 5 – Apresenta os dados para a etapa de estimação dos parâmetros do kernel RBF, bem como os resultados da classificação dos símbolos musicais manuscritos, para cada etapa da análise da validação cruzada.	59
Tabela 6 – Apresenta alguns dos erros mais comuns de classificação, identificados por suas classes e respectiva imagem. VP: Verdadeiro Positivo, FP: Falso Positivo.	60
Tabela 6 – Apresenta alguns dos erros mais comuns de classificação, identificados por suas classes e respectiva imagem. VP: Verdadeiro Positivo, FP: Falso Positivo.	61
Tabela 7 – Observa-se os resultados de três abordagens diferenciadas sobre o problema proposto neste artigo com a mesma base de dados, mesmos símbolos e método holdout.	61
Tabela 8 – Apresenta a imagem original da semínima e a imagem resultante com a região de interesse (cabeça) demarcada.	64
Tabela 9 – Apresenta os dados para a detecção e remoção das pautas.	65
Tabela 10 – Apresenta os dados para a detecção das barras de compasso e cabeças das notas.	65

Lista de abreviaturas e siglas

BOK	<i>Bag of Keypoints</i>
FN	Falso Negativo
FP	Falso Positivo
ORB	<i>Oriented FAST and Rotated BRIEF</i>
OMR	<i>Optical Music Recognition</i>
RBF	<i>Radial Basis Function</i>
SVM	<i>Support Vector Machine</i>
VC	Dimensão de Vapnik-Chervonenkis
VP	Verdadeiro Positivo
VN	Verdadeiro Negativo

Sumário

1	INTRODUÇÃO	14
1.1	Objetivo	16
1.1.1	Objetivos Específicos	16
1.2	Organização do Trabalho	16
2	TRABALHOS RELACIONADOS	18
3	FUNDAMENTAÇÃO TEÓRICA	22
3.1	Notação Musical	22
3.2	Optical Music Recognition (OMR)	26
3.3	Processamento de Imagem	28
3.3.1	Binarização	28
3.3.2	Detecção das Pautas	31
3.3.3	Operações Morfológicas	32
3.3.4	Segmentação dos Símbolos Musicais - FloodFill	33
3.4	Reconhecimento de padrões	34
3.5	ORB - Oriented FAST and Rotated BRIEF	35
3.6	BoK - (Bag of Keypoints) - Descritores Binários	38
3.7	K-Means	39
3.8	SVM - Support Vector Machine	41
3.9	Métricas de Avaliação	44
4	MATERIAIS E MÉTODOS	46
4.1	CVC-MUSCIMA (Base de Dados)	46
5	OMR PROPOSTO	49
5.1	Pré-Processamento	50
5.2	Processamento das pautas	51
5.3	Processamento dos símbolos musicais	53
5.3.1	Extração de Características - Símbolos Segmentados	54
5.4	Classificação	55
5.4.1	Resultados	58
5.5	Reconstrução da Notação Musical	62
5.6	Publicação do arquivo digital da partitura	66
6	CONCLUSÃO	68
6.1	Trabalhos Futuros	68

REFERÊNCIAS	70
--------------------------	-----------

1 Introdução

Com o passar do tempo e através dos séculos, a música e as notas musicais têm evoluído bastante, tanto em conteúdo simbólico quanto em qualidade de apresentação. Uma partitura, é uma representação formal da música, capaz de ser compreendida por qualquer pessoa que possua conhecimento na área da teoria musical, pois suas regras e padrões são estudadas mundialmente. Tal qual a matemática, as suas normas e diretrizes possibilitam a internacionalização do conhecimento.

O campo de identificação automática de símbolos manuscritos apresenta grandes desafios, devido à grande variação de como pessoas diferentes podem desenhar o mesmo símbolo. Além disso, o procedimento de identificação correta da notação musical envolve diferentes problemas, mais complexos do que o reconhecimento de texto, ([BEN-DAYAN; GILOH, 2013](#)).

As diferenças entre a notação musical e o texto vão mais longe do que a simples variação de caracteres atômicos. O texto é predominantemente unidimensional, ao passo que a música faz pleno uso do espaço bidimensional. Por exemplo, uma clave no início de uma linha, afeta notas subsequentes sobre a qual o músico, terá que aguardar até que outra clave seja encontrada, ou o fim da música seja atingido; bem como as sílabas escritas abaixo de uma linha estão associadas a notas particulares, entre outras inúmeras particularidades estruturais, ([BAINBRIDGE, 1997](#)).

A digitalização tem sido comumente usada como uma possível ferramenta para a preservação, oferecendo duplicações fáceis, distribuição e processamento digital. No entanto, é necessário um formato simbólico padronizado e legível pela máquina, das produções musicais em formato de partituras para facilitar as operações, como a busca, recuperação, análise e produção.

A transcrição manual de partituras em um formato digital adequado é realizada somente por meio de muito tempo e esforço humano. O desenvolvimento de métodos de processamento de imagem, tem contribuído para o desenvolvimento de vários algoritmos importantes, que atuam como soluções para a *Optical Music Recognition (OMR)*.

Os avanços e benefícios que a pesquisa da OMR tem realizado, também podem ser sentidos e medidos nos meios educacionais, como por exemplo: convertendo partituras em código Braille para cegos, bem como gerando uma versão personalizada de exercícios de música, conversões automáticas de tom para os mais diversos instrumentos, etc. Além disso, esta tecnologia pode ser utilizada para indexar e armazenar partituras em bancos de dados. Hoje, há uma série de bases de dados on-line contendo partituras digitais, tornando a música acessível a todos e de forma gratuita, ([JOHANSEN, 2009](#)).

O campo de pesquisa da OMR começou com (PRUSLIN, 1966) e (PREARU, 1970), que desde então vem enfrentando diversas dificuldades tanto em questões de software e hardware, nos quais se percebe de forma notória e imensurável a união entre o desenvolvimento das tecnologias de visão computacional e o avanço que esses proporcionam nas soluções da OMR. Várias outras pesquisas e estudos científicos apresentados, vem corroborar com novas soluções e métodos para os complexos problemas de conversão para um formato digital das partituras manuscritas, como: (KASSLER, 1972), que foi o revisor das duas primeiras dissertações sobre OMR citadas acima, na quais observou claramente, as incríveis possibilidades e mudanças nos paradigmas da notação musical a partir do reconhecimento automatizado, chegando mesmo a afirmar a necessidade do retorno dos musicologistas aos conceitos fundamentais da teoria musical em detrimento a tecnologia apresentada.

Seguindo, temos o trabalho de Blostein, Baird et al. (1992) que trata sobre os sistemas OMR desenvolvidos entre 1966 e 1992, mais a frente, (BAINBRIDGE; BELL, 2001), apresentaram uma metodologia genérica para a OMR utilizada por muitos pesquisadores, e ambos (HOMENDA, 2005) e (REBELO; CAPELA; CARDOSO, 2010) publicaram estudos de reconhecimento de padrões aplicados à notação musical, enquanto que (MIYAO; MARUYAMA, 2004) foram os responsáveis pela sistematização de um sistema online de reconhecimento de símbolos musicais manuscritos baseados no algoritmo do código em cadeia de (FREEMAN, 1961), utilizando-se também da programação dinâmica para o processo de reconhecimento de símbolos e construção da estrutura e semântica musical.

De acordo com Yadid-Pecht et al. (1996), a taxa de acurácia dos métodos da época era cerca de 80% a 90% de acerto. Estudo recente, realizado por (REBELO et al., 2012), mostra a atual situação das pesquisas realizadas na área de sistemas de OMR, que apresentam uma acurácia superior a 90%, mas a própria autora garante que a maioria dos sistemas testados não possuem a mesma acurácia que represente todas as nuances necessárias que um sistema OMR requer, como por exemplo: o reconhecimento unitário de símbolos pode alcançar a taxa de acerto prometido, mas quando representado como uma partitura completa, o arquivo de saída não corresponde igualmente com os acertos anteriormente apresentados.

Para tanto, a medição e comparação em termos de desempenho de diferentes algoritmos e abordagens OMR, trata-se de uma questão que já foi amplamente considerada e discutida, (BELLINI; BRUNO; NESI, 2007) e (JONES et al., 2008), conseqüentemente após avaliar objetivamente, grande parte de todo o conteúdo que já foi produzido nesse segmento, obtemos como solução para a proposta de comparar diferentes abordagens OMR, a divisão do problema em subgrupos específicos, como por exemplo: acurácia na detecção das notas musicais e as suas quase infinitas possibilidades e respectivas posições, detecção

de pautas, acidentes, ornamentos, textos manuscritos entre outras demarcações da notação musical e a reconstrução da notação musical.

1.1 Objetivo

Como referido anteriormente, é necessário definir os subgrupos de um sistema OMR a fim de possibilitar os meios necessários para a comparação e sistematização dos métodos e técnicas utilizadas, conseqüentemente, neste trabalho temos como objetivo principal a pesquisa no desenvolvimento e aprendizado para o reconhecimento e classificação de vinte classes de símbolos musicais manuscritos, que estão descritos ao decorrer desse estudo, permitindo solucionar um dos problemas primários do desenvolvimento de um sistema OMR. Tornando possível o início da produção de um software que sistematize, embora um protótipo, as principais funções de um programa OMR, gerando em sua saída um arquivo digital no formato MusicXML, que representa a partitura musical manuscrita submetida ao sistema, tornando possível o seu armazenamento, posterior edição e também viabilizando a realimentação de todo o sistema, auxiliando na inserção das novas instâncias detectadas na base de dados do classificador, e portanto, proporcionando a sistematização do refinamento da acurácia do sistema.

1.1.1 Objetivos Específicos

Para alcançar o objetivo geral pretendido, realizar-se-á os seguintes objetivos específicos:

- Desenvolver a etapa de detecção e remoção das linhas de pauta.
- Desenvolver método de segmentação de símbolos musicais.
- Desenvolver método de reconhecimento dos elementos musicais.
- Desenvolver etapa de reconstrução da notação musical.

1.2 Organização do Trabalho

Este trabalho está organizado em seis seções: Na introdução descreve-se de forma geral e resumida sobre a análise de partituras manuscritas, propiciando uma contextualização do tema, e apresentando os objetivos gerais e específicos desse estudo.

Logo adiante, no capítulo dois, temos uma busca e pesquisa sobre os trabalhos relacionados na área de reconhecimento de notas musicais manuscritas e temas correlacionados, como os algoritmos estado da arte mais utilizados nas pesquisas de análise e avaliação de documentos que serão aplicados. Estão descritos de forma detalhada no capítulo três, os

métodos e técnicas empregadas na extração, classificação e reconhecimento das partes de uma partitura.

No quarto capítulo, é apresentado o método proposto, em um modelo de etapas, onde é possível visualizar as implementações construídas a partir da metodologia concebida. Pode-se encontrar no quinto capítulo, os resultados dos experimentos realizados e no sexto capítulo, é realizada a conclusão do trabalho, na qual são considerados os principais aspectos do trabalho realizado, bem como são apresentadas propostas de trabalhos futuros.

2 Trabalhos Relacionados

Nesta seção abordaremos sobre as principais soluções e métodos mais atuais apresentados no meio científico para: reconhecer, classificar e digitalizar as partituras manuscritas, de forma a explorar as técnicas, algoritmos e abordagens propostas para esse problema multifacetado, assim como, observando o processo de obtenção da: acurácia, erro relativo, entre outras métricas relevantes ao caso desse estudo.

Iniciando a pesquisa, temos [Chanda et al. \(2014\)](#) retornando uma acurácia de 99,78% para a detecção de símbolos musicais, a priori o método proposto, constitui-se da utilização de duas metodologias para a extração de características, uma sendo o histograma de código em cadeia (Chain Code Histogram), e a outra, compondo-se do algoritmo de detecção de momentos de uma imagem, os momentos de Zernike; sendo esse, invariante em relação a transformações geométricas de rotação, translação e escala.

Cada metodologia foi aplicada sobre uma base composta de nove classes de símbolos musicais, para as quais, cada classe possui um conjunto de quatrocentas e quinze instâncias. Que foram submetidas a dois classificadores, o SVM e um classificador quadrático com modificações na sua função discriminante (MQDF). Sendo que, obtém-se os melhores resultados, quando empregado o MQDF em conjunto com o histograma de código em cadeia.

É possível observar, que a abordagem utilizada no trabalho referido, possui um resultado importante, embora os mais de 60 símbolos que a notação atual possui não tenham sido abordados, sem contar com as anotações de: andamento, dinâmica e ornamentos, por exemplo.

Enquanto que em [Oh et al. \(2017\)](#), foi realizado o reconhecimento dos símbolos musicais manuscritos segmentados, utilizando a base de dados HOMUS (??). Dois métodos para a extração das características foram empregados, primeiro o histograma de movimento angular direcional (HDA) e o histograma de movimento angular não-direcional (HUA). Na etapa de classificação foi o escolhido o SVM.

Foram selecionados vinte e três símbolos, para compor as classes do modelo, sobre as quais, têm-se um conjunto de doze mil imagens. Cujas classificações obtiveram uma acurácia de 94,19%. Um resultado alegadamente baixo, caso não se verifique que a quantidade de símbolos utilizados é suficiente para naturalmente superar a média de artigos já propostos. Portanto, é possível assim, utilizar as técnicas apresentadas como uma fonte confiável para abordagens realmente robustas de classificação para uma grande quantidade de amostras.

Por outro lado, [Rebello \(2012\)](#) apresenta uma metodologia, que descreve como: “o

trabalho que pode abrir novos caminhos de pesquisa para um novo módulo de reconhecimento automático de símbolos musicais para partituras manuscritas”. Baseado na aplicação de cinco abordagens de classificação diferenciadas, sendo esses o Hidden Markov Models (HMM), SVM, Relevance Vector Machines (RVM), Neural Networks (NN) e Nearest Neighbour (kNN).

As técnicas de classificação foram submetidas a três diferentes testes, tanto com partituras manuscritas como digitalizadas. Em todos os experimentos foram utilizadas 14 classes de símbolos descritos no artigo, totalizando 3222 símbolos manuscritos e 2521 símbolos digitais. As características extraídas foram as mesmas do trabalho de (PUGIN, 2006) e adicionalmente com as do projeto Gamera (DALITZ, 2009) e do descritor Blurred Shape Model (BSM) /citeescalera2009blurred.

Em todas as avaliações em que os classificadores foram submetidos, a melhor performance de detecção e acurácia pertence ao SVM, retornando uma precisão de 98,00% para a detecção de símbolos manuscritos. No artigo também foi abordado o uso de distorções elásticas para a melhorar o processo de classificação, embora sem sucesso. Constituindo-se de um trabalho significativo, pois permite auferir as melhores técnicas para a identificação de símbolos segmentados.

Não obstante, houve a busca por solucionar outros desafios do processo de conversão, como por exemplo: a metodologia de binarização da partitura. Importante etapa de pré-processamento, pois sem a conversão correta dos níveis de cinza, que comumente podem ter duzentos e cinquenta e seis (256) níveis, para uma versão binária, com apenas dois níveis, os estágios subsequentes serão comprometidos por não ter produzido uma imagem final possuindo somente duas categorias de cores, no caso preto e branco. O que segundo a autora pode ser refinado utilizando o método proposto denominado Novel Binarization. Ademais, a solução proposta de detecção e remoção das pautas, que consiste em encontrar o menor caminho entre as duas margens da partitura em conjunto com o algoritmo Line Tracking Height (DALITZ et al., 2008). Sendo retornado resultados com uma acurácia próxima de 97% sobre a base de dados CVC-MUSCIMA, solução superior a abordagens como a do método de (DALITZ, 2009) utilizada no framework Gamera.

Destoando do *status quo* das abordagens do OMR, em Roy, Bhunia e Pal (2017), foi proposta a identificação do autor da partitura, sem a necessidade de remoção das linhas da pauta. Utilizando, a cadeia oculta de Markov para a classificação, e o Local Gradient Histogram (LGH), para a extração das características das regiões de interesse da partitura. Os resultados obtidos revelam uma acurácia de 88,65% na identificação dos autores sobre a base CVC-MUSCIMA.

Muito embora, o próprio artigo explicita, que uma acurácia mais precisa foi obtida em outra abordagem, (GORDO; FORNÉS; VALVENY, 2013), com o mesmo propósito de identificar os autores da partitura. A abordagem está baseada, no que os pesquisadores

chamaram de *Bag of Notes*, um método modificado do *Bag of Visual Words* para a classificação das notas segmentadas, utilizando o algoritmo SIFT para a extração das características e o SVM para a classificação. A acurácia nessa nova abordagem obteve uma taxa 99,7% de acerto. Sendo tratada como uma técnica intitulada estado da arte, para a categorização de partituras pelos seus compositores, embora o artigo, não aborde nenhuma outra instância da reconstrução digital de uma partitura, como a detecção das pautas.

Como já foi possível perceber, as técnicas apresentadas sempre têm o seu interesse voltado para um nicho específico na resolução de um problema muito mais amplo, tornando o OMR, um problema considerado em aberto. O que nos leva ao trabalho pertencente a [Jr e Pecina \(2017\)](#), que apresenta mais uma técnica para detecção das posições dos símbolos musicais. A posição da nota na pauta, é definida pela posição da cabeça ou das cabeças da respectiva nota no pentagrama, sendo mais comum encontrá-las na extremidade das notas, mas isso pode variar de símbolo para símbolo, como por exemplo as claves de fá e dó, não possuem “cabeças”, mas a sua posição na pauta é fundamental, pois fornece a todas as notas daquele pentagrama ou até o surgimento de outra clave, suas respectivas posições a partir da nota de referência dessa clave. Logo, os autores optaram por aplicar as redes neurais convolucionais para a extração de características e classificação, juntamente com um método intitulado *Bounding Box Regression* para a demarcação da região de interesse da nota, no caso, a cabeça da nota. Sobre a base de dados MUSCIMA++, que contém 23352 notas segmentadas e marcadas por uma bounding box predefinida pelos autores da base. A técnica proposta foi capaz de alcançar uma acurácia de 97% de acerto para a detecção da cabeça da nota. Vale ressaltar que o método proposto não inclui a etapa imediatamente seguinte da detecção da posição da nota no pentagrama, embora apresente um resultado significativo para a sub-área.

Outra dificuldade encontrada, está na detecção das pautas e suas respectivas cinco linhas, temática pesquisada desde a década de 60, a exemplo [Fujinaga \(1988\)](#), que utilizou o método da projeção para identificação do pentagrama na partitura, e ainda permanece sendo aplicada por muitos pesquisadores, conquanto, a técnica careça de bom desempenho ao ser empregada no reconhecimento de pautas desalinhadas, curvadas ou inclinadas. Mais recentemente, temos o trabalho de [Calvo-Zaragoza, Pertusa e Oncina \(2017\)](#), que apresentou bons resultados ao aplicar as redes neurais convolucionais na identificação e remoção das pautas, acompanhado do método de correlação de pixels, conseguindo uma média de acerto de 99% para as dez variadas formas mais comuns de distorções do pentagrama. A detecção e remoção das pautas é um dos processos mais importantes do pipeline do OMR, pois as metodologias subsequentes para a detecção dos símbolos, dependem amplamente de uma nota segmentada com o mínimo de ruídos, distorções ou outras partes da partitura, no caso: as pautas.

Observa-se, que já existem técnicas desenvolvidas para cada caso citado, muito

embora, precisem ser melhoradas e testadas sobre uma maior quantidade de partituras e nos mais diversos estados de conservação. Além disso, o OMR ainda demanda de metodologias robustas para o reconhecimento completo da partitura manuscrita. A construção completa de uma versão digital, compreende muitos outros aspectos sintáticos, semânticos e estruturais que necessitam ser modelados sistematicamente na construção de um domínio morfológico para a representação da notação musical.

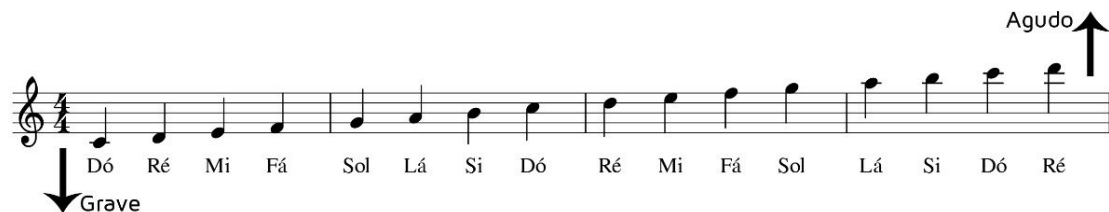
3 Fundamentação Teórica

Esta seção descreve as principais características da notação musical e um breve histórico sobre como a sua organização foi sendo modelada ao longo dos séculos até o formato que conhecemos hoje. Também apresenta os aspectos de desenvolvimento de um sistema OMR que percorre do processamento de imagens, reconhecimento de padrões, até a validação dos resultados utilizados.

3.1 Notação Musical

Como previamente discutido, o reconhecimento de notas musicais manuscritas ainda é um problema em aberto, que tem atraído a atenção de pesquisadores por sua real complexidade multifacetada, exigindo as mais diversas abordagens computacionais. Logo, na Figura 1 temos um exemplo de uma pauta em formato digital, preenchida com notas musicais na clave de sol.

Figura 1 – Exemplo de uma pauta digital com notas na clave de sol.



Fonte – Autor

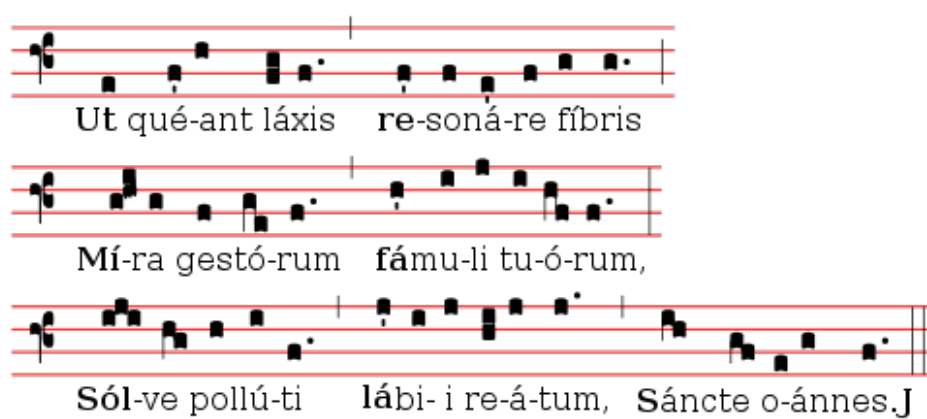
A notação musical como compreendemos, vem se transformando ao longo dos milhares de anos da história humana, as primeiras evidências conhecidas de uma notação musical padronizada vêm dos sumérios da cidade de nipur (atualmente o Iraque), encontrado na forma de um tablete em escrita cuneiforme, produzido a cerca de 2000 a.C. (KILMER, 1986), mais a frente os gregos também construíram um sistema para notação com quatro diferentes derivações que consistiam de símbolos e letras que representavam as notas em uma canção.

Entretanto, a notação que é ensinada hoje, deu seu início com o monge italiano por nome de Guido d'Arezzo (OTTEN, 1910) no período compreendido de 992 d.C. – 1050 d.C., entre as muitas reformas feitas, temos a criação da notação absoluta das alturas (onde para cada nota, temos uma posição na pauta), bem como o desenvolvimento do

tetragrama, momento no qual podemos citar como as primeiras partituras (BENNETT, 1986) que permitiam aos músicos grafarem as suas composições.

Além disso, o monge beneditino também é famoso por cunhar os nomes das notas musicais que são facilmente reconhecidas hoje, (Dó, Ré, Mi, Fá, Sol, Lá e Si). Os nomes foram extraídos das sílabas iniciais do Hino a São João Batista, (GROUT; PALISCA, 1988), como pode ser visto na Figura 2.

Figura 2 – Exemplo de um tetragrama, Hino a São João Batista, de onde foi extraído o nome das notas que conhecemos hoje.



Fonte – (KOLOMOETS, 2017)

Existem também outros modos de apresentar um conjunto de notas a ser reproduzido, essas são as cifras. As cifras são as representações gráficas dos acordes, que por sua vez são formados por um agrupamento de três ou mais notas que necessitam ser tocadas simultaneamente. As inúmeras possibilidades de combinarmos as notas e por consequente criarmos novos acordes, produz igualmente a inevitabilidade de denotá-las graficamente em um formato simbólico que chamamos de cifras. Conforme CHEDIAK (1986), “as cifras são os símbolos criados para representar o acorde de maneira prática”. A constituição das cifras é denotada por letras, que variam de A a G, números e sinais como por exemplo o sustenido (#). Abaixo, temos um exemplo de partitura em cifras do hino nacional brasileiro para violão. As cifras são um exemplo prático, da evolução da notação musical, se adaptando conforme as necessidades e mudanças culturais, ou mesmo sociais, guiando-se pelo movimento contínuo das realidades e expectativas humanas.

Consequentemente, podemos afirmar que com o passar das eras a notação musical ocidental foi sofrendo inúmeras evoluções até chegar no padrão ocidental reconhecido internacionalmente. As diversas alterações realizadas, procuram possibilitar mais dinamicidade ao aportar as novas formas de criar música, valorizando as interpretações diferenciadas bem como a adição de novos instrumentos ao mundo da música. Quando pensamos em

Figura 3 – Seção cifrada do hino nacional brasileiro.

Hino Nacional Brasileiro
Estudos para violão
Prof. Juarez Barcellos

Letra: Joaquim Manoel Bandeira Música: Osório Duque Estrada

The musical score is presented in three staves. The first staff begins with a treble clef and a 4/4 time signature. It contains several measures with trills (tr) and chords (F, C° C7). The second staff starts at measure 5 and includes chords (Gm, Gm(7M), Gm7 Gm6, C7, Fm F). The third staff starts at measure 9 and includes chords (F, F, Gm, Gm, Am, Am). The notation includes various rhythmic values, trills, and chord symbols.

Fonte – Adaptado de (BARCELLOS, 2017)

uma partitura, talvez erroneamente acreditemos que todos os músicos a interpretarão do mesmo modo, ainda que atualmente a notação musical tenha se desenvolvido o suficiente para permitir tal feito, o músico ou maestro ainda possui a capacidade sob certos aspectos de imprimir as suas peculiaridades e desejos em sua apresentação, tornando-a única. Deixando explícito, o interminável aperfeiçoamento da música que por sua vez representa o inextinguível avanço humano.

O conjunto de cinco linhas para a pauta musical, também chamada de pentagrama, foi um importante marco na transformação da notação musical, conquanto, o seu conjunto de pautas e todos os outros símbolos musicais formam a partitura, um exemplo pode ser visualizado na Figura 4.

Na Figura 5, é possível visualizar a mesma partitura, mas em sua versão manuscrita, o que nos deixa evidente a substancial dificuldade que encontramos ao desenvolver um sistema capaz de automatizar a conversão digital de partituras manuscritas. É possível notar, de forma a ater-se somente ao aspecto superficial do trabalho, que existe a necessidade de termos que lidar primariamente, com: ruídos, manchas, borrões, grafias diminutas e informações textuais que também são importantes no momento da execução musical e que precisam ser compreendidas e reinseridas em sua versão digital.

Figura 4 – Um exemplo da notação digital atual, Prelúdio, Op. 28, No. 7, Frédéric Chopin.

Prelude
Op. 28, No. 7

Frederic Chopin

Piano

p dolce

con pedale

mp

mp

rit. e dim. - - - pp

Fonte – Adaptado de (KLINDWORTH, 2017)

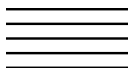
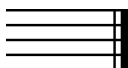

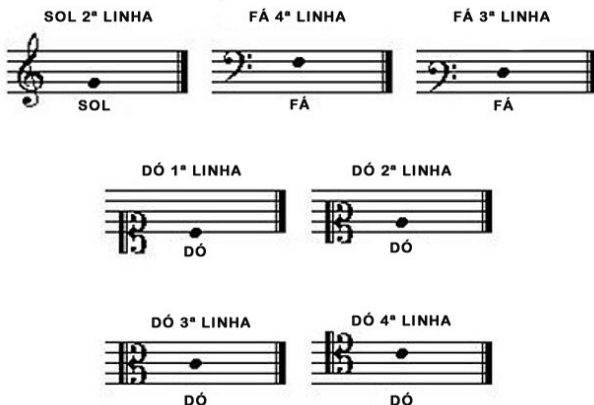
Figura 5 – Versão manuscrita da Sonata para piano e violoncelo, Op. 65, Frédéric Chopin

Fonte – Adaptado de (CHOPIN, 2017)

Como pode-se observar, a partitura é um documento que possui uma estrutura pré-definida ou seja formalizada, a partir dela podemos definir alguns padrões, podemos ver alguns exemplos na Tabela 1

Foi possível observar, que a notação musical possui uma estrutura e regras bem definidas, o que nos permite criar, desenvolver e reproduzir música. As cifras, assim

Tabela 1 – Algumas das principais características da notação musical.

Notação	Descrição
Toda pauta é composta por cinco linhas e quatro espaços.	
A barra final demarca o fim de uma composição.	
Acidentes modificam a altura das notas à sua direita e de todas as notas na mesma posição na pauta até o final do compasso corrente.	
As claves definem a faixa de altura ou a tessitura que a pauta representa.	 <p>SOL 2ª LINHA FÁ 4ª LINHA FÁ 3ª LINHA</p> <p>SOL FÁ FÁ</p> <p>DÓ 1ª LINHA DÓ 2ª LINHA</p> <p>DÓ DÓ</p> <p>DÓ 3ª LINHA DÓ 4ª LINHA</p> <p>DÓ DÓ</p>

como as partituras, modelam simbolicamente as interpretações musicais, muito embora saibamos que as partituras forneçam mais dados e informações, e por esse motivo seja mais amplamente usada para apresentar uma composição musical em sua totalidade. Pois, as nossas mais primitivas canções, bem como as mais sofisticadas orquestrações, são o registro físico do desenvolvimento da sociedade humana, elucidam as nossas crenças, regozijos, medos, anseios, lutas, expectativas, modos de vida. Estão todas representadas na incrível história da música por meio das partituras.

3.2 Optical Music Recognition (OMR)

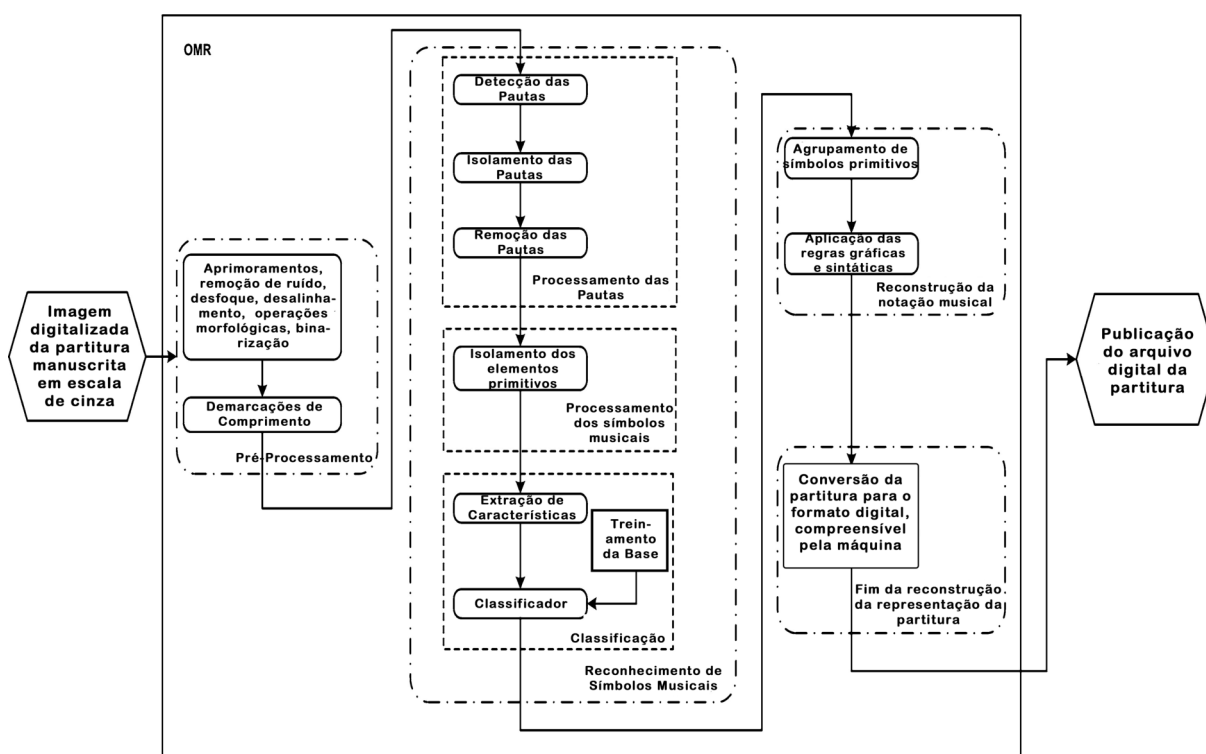
Conforme os anos, e as décadas passam, muitos músicos ainda mantêm a tradição de escrever suas partituras de forma tradicional, com caneta e papel, alguns por terem vivido em uma época onde a tecnologia estava em seus primórdios e assim sendo, nunca conseguiram realizar essa transição, ou mesmo por uma questão de gosto e praticidade ao escrever sobre a folha física. Mas a questão importante que permanece, nesse caso, é a de

que milhares e milhares de criações musicais, que não deixam de ser patrimônio artístico que inspiraram e ainda influenciam milhões, podem se perder com o tempo (REBELO et al., 2012). Diversos métodos e abordagens foram propostos ao longo dos anos, mas nenhum que solucionasse efetivamente a automatização do processo de conversão de partituras manuscritas.

Em Jones et al. (2008), foi apresentado um estudo mais completo, que incluiu a digitalização, o reconhecimento e conversão de um sistema OMR. Temos também artigos como (REBELO et al., 2012) e (FORNÉS; SÁNCHEZ, 2014) que descrevem de forma geral as etapas de um sistema OMR como: pré-processamento, reconhecimento e remoção das pautas, segmentação dos símbolos, reconhecimento dos símbolos e reconstrução da notação musical.

Na Figura 6, é possível visualizar graficamente a arquitetura generalista que um sistema OMR deve apresentar para conseguir solucionar ao menos os problemas mais conhecidos.

Figura 6 – Arquitetura típica de um sistema de processamento OMR



Fonte – Adaptado de (REBELO, 2012)

Inicialmente, temos a etapa responsável pelo pré-processamento da imagem, na qual são realizados processos de redução de ruídos e manchas, bem como ajustes de escala e rotação. E também a binarização da mesma, permitindo a distinção entre o fundo da imagem e o seu conteúdo. Logo em seguida, temos o reconhecimento dos símbolos extraídos,

que passaram por diversos processos, que começam desde a detecção e remoção da pauta, permitindo assim que o símbolo seja extraído sem partes da pauta, o que facilita o processo consequente de classificação do mesmo. E prosseguimos com a criação da estrutura da notação musical, que por sua vez também inclui inúmeros processos de análise estrutural e sintática, e por fim o sistema nos retorna a imagem digital em sua representação final.

Vejamos com mais detalhes as principais etapas que um sistema OMR precisa desenvolver, nas seções a seguir.

3.3 Processamento de Imagem

Quando falamos sobre processamento de imagem, um dos primeiros aspectos a ser considerado, é a resolução do arquivo que será analisado. É notório e patente, que o impacto causado na utilização de imagens com resoluções entre 300dpi e 600dpi, é refletido positivamente em toda a cadeia subsequente do sistema OMR. Muitas das partituras manuscritas disponíveis, estão desgastadas pelo tempo e armazenamento inadequado, sendo necessária a digitalização do material em máxima resolução disponível, pois além de propiciar um registro de alta qualidade, muitas partituras também são consideradas artefatos históricos. Onde cada detalhe, denota um peso importante no sucesso final. Portanto, essa faixa de resolução é utilizada pela maioria dos autores em seus métodos, como em (DALITZ et al., 2008), (RILEY; FUJINAGA, 2003), refletindo diretamente em todo o processo de segmentação, classificação e conversão.

Por outro lado, a análise de documentos, não se limita a resolução do arquivo, também recorre a outros métodos de pré-processamento. No nosso caso de estudo, a binarização é um passo fundamental para o processo de reconhecimento. Como Rebelo et al. (2012), faz questão de ressaltar, quase todos os sistemas OMR começam com um processo de binarização.

3.3.1 Binarização

O método da binarização, pode ser compreendido pela redução da escala de cores da imagem apresentada para apenas duas cores, preto e branco. No caso das partituras manuscritas, o objetivo do processo é diferenciar o fundo da partitura, dos símbolos e notação musical. A binarização é um problema muito conhecido no campo da visão computacional, as mais diferentes abordagens já foram propostas como em (TRIER; TAXT, 1995), (KAVALLIERATOU; STATHIS, 2006). Uma das técnicas mais utilizadas, é conhecida como método baseado em limiar ou originalmente, Threshold Based method, que converte a imagem em tons de cinza para uma escala binária. Entretanto, o método necessita que seja informado o limiar global a ser calculado, inviabilizando a automação, principalmente sobre imagens que tenham sofrido danos, como comumente encontrado em

partituras manuscritas, pois o limiar conseqüentemente vai variar entre as partituras e até mesmo localmente já que o valor de limite é definido para toda a imagem da partitura, nesse caso.

No entanto, uma versão capaz de calcular automaticamente o valor do *threshold* foi proposta, originalmente chamado de Adaptive Thresholding Based (SAUVOLA et al., 1997), o algoritmo realiza o cálculo do limiar sobre regiões menores, obtendo diferentes valores de limite e melhorando notadamente a acurácia, pois desse modo o limiar é customizado para cada região conforme a sua luminância.

Outro algoritmo de binarização, também muito conhecido e utilizado, é o método de Otsu's, (OTSU, 1979). Para compreendermos esse algoritmo, precisamos conhecer as imagens bimodais.

Essas imagens são identificadas por possuírem dois picos, ou duas regiões de luminância com maior incidência de pixels. Logo, para calcularmos o valor do limiar da imagem, precisaríamos procurar por dois picos na imagem, representadas por valores de pixel do primeiro plano e do fundo, e então calcular o valor médio da região entre eles. Sendo esse, exatamente o processo de detecção que a técnica de Otsu's propõe.

O cálculo está fundamentado no agrupamento da contagem do histograma. Se tomarmos os valores dos pixels como $i = 1, 2, \dots, L$; logo o conjunto da contagem do histograma i é representado como n_i , e a frequência relativa compreendida como $p_i = n_i/N$, onde N é o número de pixels da imagem. O trabalho de calcular o limiar da imagem, está baseado em separar os pixels da imagem em duas classes distintas. C_0 pertence ao agrupamento dos pixels com valores $[1, \dots, K]$ e C_1 é o conjunto dos pixels com valores entre $[K+1, \dots, L]$.

Logo as probabilidades gerais das classes w_0 e w_1 são:

$$w_0 = \sum_{i=1}^k p_i = w(k) \quad (3.1)$$

$$w_1 = \sum_{i=k+1}^L p_i = 1 - w_0(k) \quad (3.2)$$

E os valores médios dos pixels de C_0 e C_1 podem ser obtidos utilizando:

$$\mu_0 = \frac{\sum_{i=1}^k ip_i}{w_0} = \frac{\mu(k)}{w(k)} \quad (3.3)$$

$$\mu_1 = \frac{\sum_{i=k+1}^L ip_i}{w_1} = \frac{\mu_T - \mu(k)}{1 - w(k)} \quad (3.4)$$

Onde,

$$\mu(k) = \sum_{i=1}^k ip_i \quad (3.5)$$

$$\mu_T = \sum_{i=1}^k ip_i \quad (3.6)$$

A variância das classes pode ser obtida por meio dos operadores σ_0^2 e σ_1^2 , descritos abaixo:

$$\sigma_0^2 = \sum_{i=1}^k \frac{(i - \mu_0)^2 p_i}{w_0} \quad (3.7)$$

$$\sigma_1^2 = \sum_{i=k+1}^L \frac{(i - \mu_1)^2 p_i}{w_1} \quad (3.8)$$

O método de Otsu emprega três abordagens para diferenciar as classes: variância interna (γ), variância entre as classes (k) e a variância total (δ). Que podem ser obtidas por:

$$\gamma = \sigma_B^2 \quad (3.9)$$

$$k = \frac{\sigma_T^2}{\sigma_w^2} \quad (3.10)$$

$$\delta = \frac{\sigma_B^2}{\sigma_T^2} \quad (3.11)$$

Nas quais, temos que:

$$\sigma_w^2 = w_0 \sigma_0^2 + w_1 \sigma_1^2 \quad (3.12)$$

$$\sigma_B^2 = w_0 (\mu_0 - \mu_T)^2 + w_1 (\mu_1 - \mu_T)^2 = w_0 w_1 (\mu_1 - \mu_0)^2 \quad (3.13)$$

Otsu's afirma que maximizar um dos critérios acima, significa maximizar os outros implicitamente. Além disso, se maximizarmos δ , estaremos maximizando σ_B^2 , que pode ser reescrito em termos do limite selecionado, (k):

$$\sigma_B^2(k) = \frac{w_0 [\mu_T w(k) - \mu(k)]^2}{w_k [1 - w(k)]} \quad (3.14)$$

A equação acima representa o cerne do algoritmo de Otsu. Onde todos os valores possíveis dos limiares encontrados em σ_B^2 , são calculados e escolhemos, por fim, o valor do limite que maximiza a função central.

Logo, podemos garantir por meio desse método a automatização do processo de binarização e por consequente, a extração das características da partitura e suas propriedades, sem as particularidades do fundo da partitura. Portanto, adquirimos assim, uma imagem binarizada, composta simplesmente de 0's e 255's, possibilitando o prosseguimento das operações de pré-processamento como a identificação das pautas.

3.3.2 Detecção das Pautas

A detecção e segmentação das pautas é um problema recorrente e completamente presente no processo de identificação das partituras, como citado anteriormente, apenas poucos pesquisadores propõe a não remoção das linhas de pauta para o processo de identificação e análise, como em [Roy, Bhunia e Pal \(2017\)](#), muito embora os pesquisadores [Gordo, Fornés e Valveny \(2013\)](#), tenham proposto a remoção das pautas para o processo de identificação do autor da partitura e tenham obtido melhores resultados.

O processo de remoção das pautas, também permeia outros âmbitos do processo de extração de características, pois permite a segmentação e análise dos símbolos presentes na partitura sem adição de outras propriedades que podem interferir no processo de reconhecimento. Um dos algoritmos mais utilizados, como nos esclarece a pesquisadora ([REBELO, 2012](#)), pertence a Christoph Dalitz, ([DALITZ et al., 2008](#)), também utilizado no Framework Gamera, o qual têm os seus valores de espessura e distância entre linhas estimados pela técnica de ([FUJINAGA, 2004](#)). A metodologia de Dalitz apresenta um algoritmo para a pesquisa e identificação de segmentos de pauta, que nos mostram também técnicas não somente para assinalar a conexão entre esses segmentos, mas bem como a incorporação de superposições entre esses segmentos de forma a interligar tanto horizontalmente quanto verticalmente as linhas. Formando assim um grafo para posterior análise. Logo abaixo temos uma descrição generalista da metodologia de Dalitz:

1. Adicione links verticais entre os segmentos de pauta com uma distância aproximada por: $\text{linhapauta_altura} + \text{espaçopauta_altura}$.
2. Adicione links horizontais entre os segmentos de pauta adjacentes, que possivelmente pertencem à mesma linha.
3. Particionar o grafo resultante em subgrafos conectados; onde para cada subgrafo, que seja suficientemente largo e alto, vai corresponder a uma pauta.

4. Todos os segmentos de pauta, dentro de um sistema, são rotulados como pertencentes a uma determinada linha de pauta. Segmentos da mesma linha na mesma posição horizontal são mesclados em um mesmo segmento.
5. Devido às linhas suplementares, ligaduras e barras da partitura, alguns subgrafos possuirão muitas linhas de pauta. Para reduzi-las a um número predefinido de linhas por pauta (tipicamente cinco para a notação moderna, quatro para o canto e seis para a tablatura), as linhas são posteriormente removidas até o número conhecido de linhas de pauta.

E finalmente utilizando o algoritmo abaixo, Dalitz propõe encontrar os segmentos de pauta:

1. Extrair as linhas horizontais com mais de 60% de pixels pretos dentro de uma região de largura representada por: `espaçopauta_altura`.
2. Os resultados das linhas extraídas, são então processados e por consequente extraído a posição vertical central das mesmas, realizando um processo de esqueletonização. Para linhas que possuem uma altura maior que $2 * \text{linhapauta_altura}$, dois pontos serão extraídos e armazenados.
3. Logo, os segmentos retornados que possuem uma largura maior $2 * \text{linhapauta_altura}$, são consideradas como segmentos de pauta.

Utilizando os segmentos de pautas, e posterior identificação das linhas e conseqüentemente as pautas da partitura, o processo de identificação foi concluído com êxito e o sistema agora possui as posições das pautas e suas respectivas linhas, permitindo a localização das claves, bem como dos símbolos musicais nas pautas, proporcionando a base primordial de informações para a compreensão da partitura.

3.3.3 Operações Morfológicas

As imagens também são processadas sobre os métodos conhecidos como: operações morfológicas. Essas operações, conforme [Haralick, Sternberg e Zhuang \(1987\)](#), podem ser descritas de maneira a propor: “Operações orientadas à forma, que simplificam os dados da imagem, preservando suas características de estrutura essenciais e eliminando irrelevâncias”. Foram definidas pelos matemáticos ([MATHERON, 1967](#)) e ([SERRA, 1982](#)). Elas são conhecidas por sua capacidade de extrair os componentes da imagem, que por sua vez, são úteis para realizar a supressão de ruído, análise de textura, análise de formas, detecção de bordas, esqueletonização, entre inúmeras outras abordagens. A teoria dos conjuntos descreve a representação que é aplicada sobre a morfologia matemática,

portanto, é capaz de ser utilizada em imagens binárias sobre as quais geralmente as operações morfológicas são utilizadas. Onde, para um ponto que está definido sobre o grupo de pixels do primeiro plano ou do plano de fundo as operações de conjunto, que nos são amplamente conhecidas, como: (união, intersecção, complemento), são aplicadas sobre ele.

Como observamos, uma filtragem morfológica, é constituída de transformações aplicadas sobre a imagem por meio de elemento estruturante. Que por sua vez, atua como uma sonda de busca, que seja sensível à informação geométrica, trabalhando no sentido de identificar formas da imagem, que são semelhantes em estrutura e tamanho com o elemento estruturante, com a finalidade de preservar suas características, enquanto que outras regiões são extraídas ou removidas.

A dilatação, possui a característica de ampliar as áreas dos pixels do primeiro plano, essas áreas crescem em tamanho, enquanto que as partes das áreas internas que representam o fundo da imagem, diminuem de dimensão. A erosão, por sua vez, propõe a redução de áreas de borda e, portanto, a remoção dos limites dos pixels do primeiro plano. Logo, as áreas dos pixels do primeiro plano diminuem e as regiões do segundo plano dessas áreas tornam-se maiores.

As operações morfológicas, são recursos sempre presentes na base da análise de documentos (COMER; DELP, 1999), conseqüentemente, no estudo sobre as partituras manuscritas, a técnica encontra inúmeros meios para ser aplicada, como por exemplo, temos Ng (1999), que propôs a utilização da morfologia matemática na extração e segmentação das partes da partitura, utilizando a técnica da esqueletonização e detecção de bordas. Também, encontramos a técnica sendo utilizada como ferramenta para a detecção das cabeças das notas. Os autores, Fornés, Lladós e Sánchez (2005), apresentam a técnica da operação morfológica de dilatação na detecção das cabeças das notas, utilizando uma elipse como elemento estruturante, bem como um elemento estruturante vertical na localização das linhas verticais.

3.3.4 Segmentação dos Símbolos Musicais - FloodFill

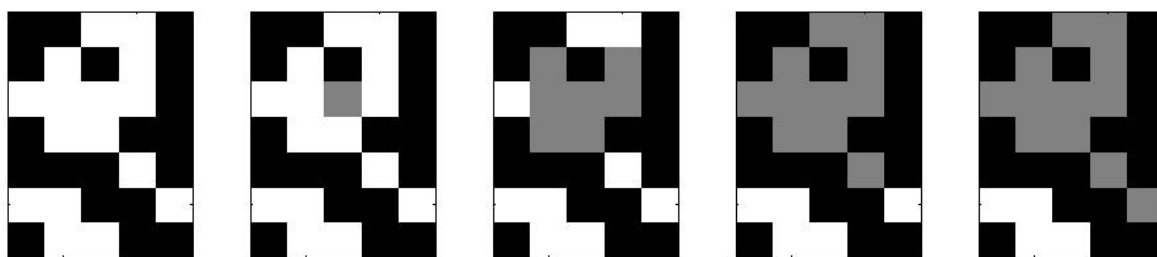
Outra importante etapa, consiste na extração unitária dos símbolos musicais, que implica na identificação de componentes conectados. No caso das partituras, tanto manuscritas quanto digitais, após realizar as etapas de binarização da imagem e remoção das pautas, obtemos como resultado, um conjunto reduzido de símbolos musicais com componentes interligados, pois uma das estruturas que mais resultam em conexão entre símbolos, que está compreendido na forma de linhas de pauta, foi removida.

Os principais algoritmos utilizados para a extração de regiões que estejam conectadas em imagens binárias, consiste em traçar contornos (*Contour Tracing*) (SUZUKI et al.,

1985) ou o algoritmo denominado *Flood Fill* que pode ser aplicado sequencialmente a todos os pixels do primeiro plano da imagem (GRANA; BORGHESANI; CUCCHIARA, 2009).

A abordagem empregada nesse trabalho, se valeu do algoritmo Flood Fill (TORBERT, 2016), que consiste em determinar a área conectada de uma determinada célula dentro de um *array* multi-dimensional, consideremos um grafo, sendo representado por uma matriz, onde cada célula da matriz funciona como um vértice conectado a outras células à direita, esquerda, abaixo e acima, bem como nas diagonais, no caso para o modo de 8 conexões. Um exemplo do funcionamento do algoritmo pode ser observado na Figura 7.

Figura 7 – Ilustração do funcionamento do algoritmo Flood Fill



Fonte – Fonte (NOSAL, 2008)

Portanto, considerando o exposto, verificamos que a etapa de processamento de imagens é fundamental para detecção e análise de partituras manuscritas, pois conforme analisado, as técnicas apresentadas, interferem diretamente na qualidade dos resultados obtidos, seja por meio da binarização, operações morfológicas ou técnicas em conjunto para a segmentação de símbolos. Desse modo, torna-se manifesto as inúmeras dificuldades que precisam ser sanadas para a análise de partituras manuscritas, que possui a análise de documentos como aliada fundamentas nesse processo.

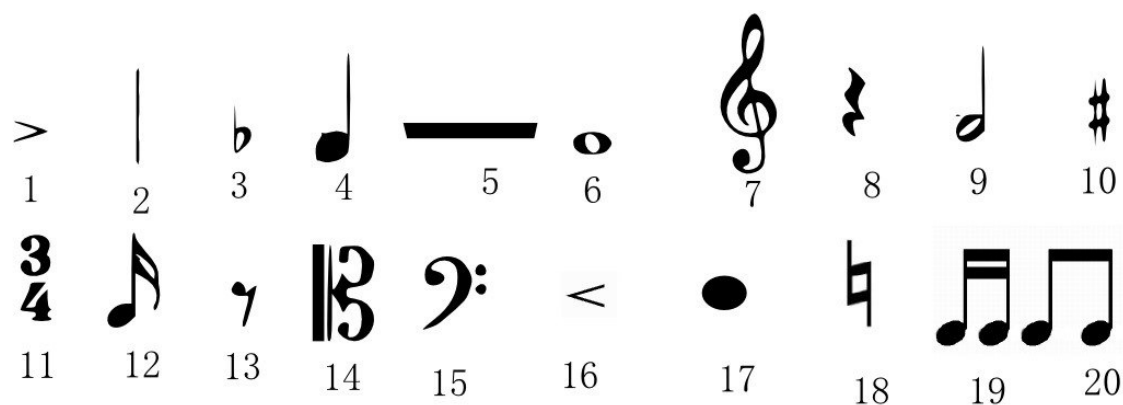
3.4 Reconhecimento de padrões

O reconhecimento de padrões em visão computacional, constitui-se do processo de outorgar um rótulo ou identificar um objeto em categorias ou classes. Como afirmou o pesquisador, Christopher M. Bishop, a respeito do reconhecimento de padrões:

“O campo de reconhecimento de padrões, está interessado na descoberta automática de regularidades em dados, através do uso de algoritmos computacionais, e com o uso dessas regularidades para tomadas de ações, como: classificação dos dados em diferentes categorias” (BISHOP, 2006).

Logo, o processo de reconhecimento de padrões deve ser capaz de identificar e categorizar os símbolos extraídos da partitura manuscrita, em conformidade com as categorias descritas na Figura 8. Realizando a detecção das respectivas classes, para cada segmento da partitura, que representa um símbolo musical, que tenha sido submetida ao sistema.

Figura 8 – Todas as classes com seus respectivos símbolos utilizados nessa pesquisa.



Fonte – Autor

Como resultado, se faz necessário a extração de características dos símbolos segmentados da partitura, para posterior análise do sistema de classificação e reconhecimento.

Portanto, em nosso objetivo dispomos quase que invariavelmente de duas cores, uma do papel ou plano de fundo e outra da partitura descrita ou primeiro plano. Fazendo-se necessário a captura dos símbolos musicais, para a extração das características, de modo a apresentarem suas propriedades possuindo atributos que sejam: invariantes a escala, aspecto e com tratamento para manchas ou outras marcas deixadas pelo tempo e modo conservação.

Possibilitando a etapa de reconhecimento, a identificação de símbolos que estejam em diferentes ângulos de apresentação, mesmo possuindo diferentes tamanhos e formas. Assim, foi escolhido o algoritmo capaz de lidar com ruídos, e invariante a escala e rotação, denominado: ORB (Oriented FAST and Rotated BRIEF) para essa pesquisa. A seguir, uma descrição do algoritmo utilizado.

3.5 ORB - Oriented FAST and Rotated BRIEF

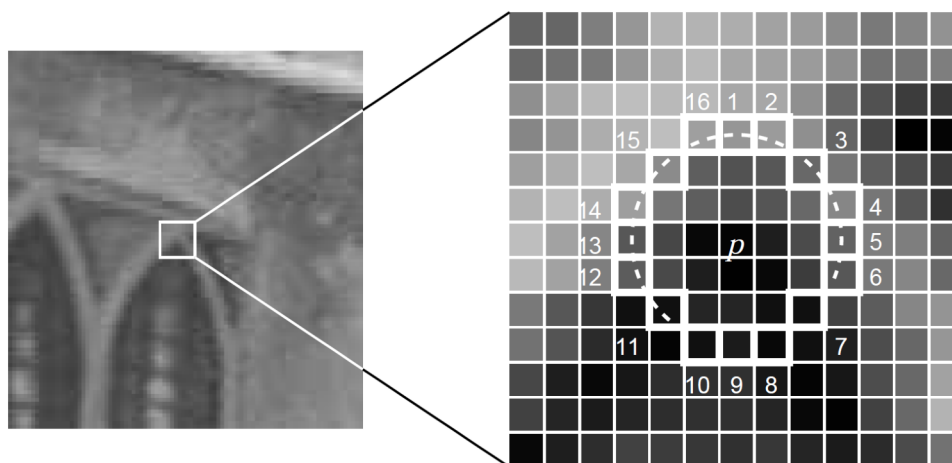
De acordo com, [Ruble et al. \(2011\)](#) foi proposto um descritor binário baseado no BRIEF ([CALONDER et al., 2010](#)) e FAST ([ROSTEN; PORTER; DRUMMOND,](#)

2010), chamado ORB, que é invariante a rotação e capaz de tratar o ruído. Os autores demonstram através de experimentos, que o ORB é até duas vezes mais rápido do que o SIFT (LOWE, 1999), durante a execução, bem como em muitas situações. Segundo os autores, a eficiência foi testada em várias aplicações do mundo real, incluindo a detecção de objetos utilizando patch-tracking em smartphone's.

Logo, o ORB é basicamente uma fusão do detector de KeyPoints - FAST e o descritor binário BRIEF. Com modificações para melhorar o desempenho e acurácia do método. Primeiro, usa-se o FAST para encontrar os KeyPoints, que seriam os pontos de interesse da imagem, possuindo o maior número de informações locais, que se espera idealmente ser encontrado também em outras imagens diferentes.

O FAST (Features from Accelerated Segment Test) busca essencialmente por cantos em imagens como sendo regiões de interesse, mas diferente dos algoritmos que utilizam diferenças de gaussianas (DoG), o FAST realiza a pesquisa por meio de um círculo de 16 pixels para classificar se um ponto candidato p é realmente um canto. Logo, se um conjunto de N pixels contíguos no círculo, estiverem mais brilhantes ou escuros do que a intensidade do pixel p (candidato), somados a um limiar t , em relação a intensidade do valor do limite de p daquele pixel, então p será classificado como um canto. Na Figura 9 é possível visualizar o ponto candidato p e o círculo de teste ao seu redor.

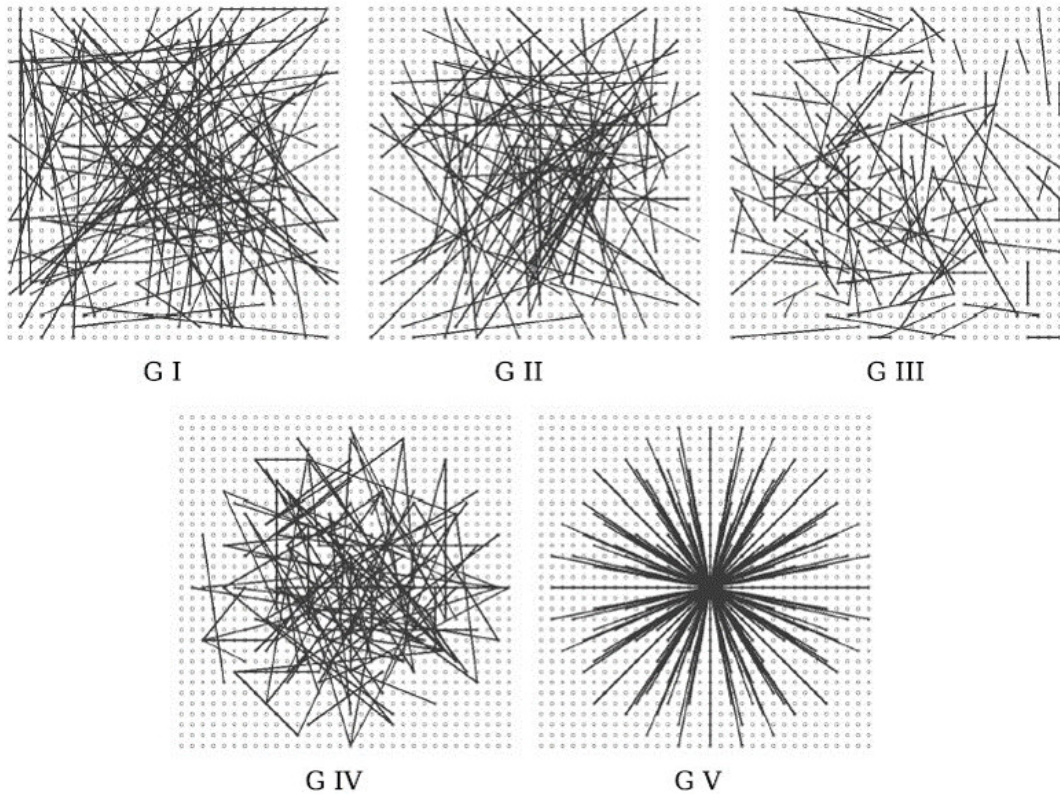
Figura 9 – Detector FAST: Exemplo do ponto de interesse p , e os 16 pixels no círculo de teste.



Fonte – (ROSTEN; PORTER; DRUMMOND, 2010)

Quanto ao BRIEF, aplicado em seguida, para extrair os descritores das regiões de interesse, este incorpora apenas a informação presente na localização do pixel retornado, gerando randomicamente padrões de amostragem, como pode ser visto na Figura 10, para criar a sequência binária que representa uma das regiões ao redor do KeyPoint.

Figura 10 – BRIEF: Exemplo de cinco padrões de amostragem.



Fonte – (CALONDER et al., 2010)

No ORB a informação de orientação que falta no FAST é complementada com Rosin's corner intensity, (ROSIN, 1999).

As imagens nas quais estamos interessados possuem duas dimensões, logo, precisamos de duas variáveis independentes: x e y para acessá-las. Onde p e q representam a ordem dos momentos.

O momento M do patch (região de interesse) I pode ser calculado como:

$$m_{pq} = \sum_{x,y} x^p y^q I_p(x, y) \quad (3.15)$$

Podemos calcular o centroide C (Centro de Gravidade do Objeto) como:

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad (3.16)$$

Pode-se definir a orientação relativa do patch como:

$$w = \text{atan2}(m_{01}, m_{10}) \quad (3.17)$$

O descriptor patch foi fornecido a partir do operador BRIEF, uma representação de cadeia de bit's construído a partir de um conjunto de testes de intensidade binários. Um teste binário t pode ser realizado como:

$$t(I_p, x, y) = \begin{cases} 1 & : I_p(x) < I_p(y) \\ 0 & : I_p(x) \geq I_p(y) \end{cases} \quad (3.18)$$

Onde, $I_p(x)$ é a intensidade do ponto x , e $I_p(y)$ é a intensidade do ponto y . O vetor de características do descriptor BRIEF é definida como um vetor de testes binários, de tamanho n (128, 256 bits):

$$b(I_p) = \sum_{1 \leq i \leq n} 2^{i-1} t(I_p, x_i, y_i) \quad (3.19)$$

Uma vez que tenhamos calculado a orientação do patch, pode-se realizar uma rotação canônica e, em seguida, calcular o descriptor, obtendo-se assim alguns momentos de invariância a rotação.

Após o processo de extração das características, os dados binários retornados pelo descriptor ORB, são submetidos a um modelo de dados denominado: bag of keypoints, ou em tradução literal: saco de pontos-chave, que se constitui de uma técnica de agrupamento de keypoints proposta por (CSURKA et al., 2004), utilizado em uma ampla gama de aplicações tanto genéricas como específicas para categorização visual.

3.6 BoK - (Bag of Keypoints) - Descritores Binários

A técnica de classificação do bag of keypoint, está baseada no modelo BoW (Bag of Words), que trata uma imagem como um repositório de "palavras-código", que consiste essencialmente em um conjunto de descritores locais independentes. Essas características são extraídas por descritores como o ORB, por meio de amostragens aleatórias, segmentação, ou grade regular.

Enquanto que características baseadas em histograma estão prontas para serem usadas na classificação de imagens ou tarefas de recuperação, descritores locais requerem um passo de quantização adicional para ser transformado em características da imagem total.

Logo, os pesquisadores Csurka et al. (2004), apresentam algumas etapas principais para o método proposto:

1. Detecção e descrição de fragmentos da imagem.

2. Atribuição dos descritores dos fragmentos a um conjunto de clusters pré-determinados (ou seja, o vocabulário) utilizando um algoritmo de quantização vetorial.
3. Construção de um agrupamento de pontos-chave (Keypoints), que enumera o número de fragmentos atribuídos a cada cluster ou grupo, produzindo dessa forma um dicionário de características.
4. Aplicação de um classificador com suporte a múltiplas classes sobre o dicionário definido na etapa anterior, sendo esse apresentado como um vetor de características e, assim, retornando à indicação de qual categoria ou categorias à imagem está atribuída.

Assim sendo, a abordagem clássica é empregar o algoritmo K-means, utilizando a distância euclidiana entre vetores de características, e isso provou ser eficaz, mesmo que computacionalmente custoso, durante a fase de treinamento, para a construção do dicionário.

3.7 K-Means

O algoritmo de classificação K-means, foi empregado primeiramente por (MACQUEEN *et al.*, 1967). A técnica é categorizada como um método de aprendizado não supervisionado, uma solução relativamente simples para o bem conhecido problema de agrupamento.

A ideia, presume a classificação de um certo conjunto de dados, a partir de um número de clusters, definidos a priori. O procedimento segue com a definição de k centroides, um para cada cluster.

Um problema encontrado nessa etapa, é a definição das regiões onde cada centroide será definido, pois como o resultado será afetado substancialmente em relação ao local onde o centroide foi inserido, é preciso escolher com cuidado essas regiões, sendo tomadas regularmente a abordagem de colocá-los aleatoriamente sendo mais distantes uns dos outros. Refletindo também no resultado final, pois regiões de abordagens diferentes resultam em soluções diferentes.

O próximo passo, é tomar cada ponto pertencente a um determinado conjunto de dados e associá-lo ao centroide mais próximo.

Esse processo deve ser repetido até que nenhum ponto ainda esteja pendente, finalizando assim a primeira etapa com os agrupamentos iniciais concluídos.

A partir dessa etapa, será necessário, repetir o processo de calcular os centroides, mas agora eles estão definidos como os baricentros dos cluster calculados na etapa anterior.

Logo, após calculados as suas novas posições, o processo de ajuste dos pontos de agrupamento é reiniciado procurando o novo centroide mais próximo.

Essa etapa deve ser repetida até que os novos centroides que forem sendo criados, não mudem mais a sua localização, e conseqüentemente não haja novas alterações, completando a metodologia de classificação.

O K-means pertence à uma família de algoritmos onde os seus centroides possuem mobilidade e permissão para se moverem, ou seja, a cada iteração o centro (ou centróide) do cluster, se move moderadamente para minimizar a função objetivo.

Que pode ser descrita como:

$$J = \sum_{j=1}^K \sum_{i=1}^n X_i - C_j^2 \quad (3.20)$$

Onde, $X_i - C_j^2$, é uma medida de distância escolhida entre um ponto de dados X_i e o centro do cluster C_j , portanto representa a distância dos n pontos de dados para o seus respectivos centros de cluster.

Conforme afirma ([MELCHIORRE et al., 2008](#)), podemos resumir o algoritmo em quatro etapas:

1. Posicionar os pontos K no espaço, delimitado pela região dos objetos que estão sendo agrupados. Os pontos K representam os centroides do grupo inicial.
2. Atribuir cada objeto ao grupo que possui o centróide mais próximo.
3. Verificar se todos os objetos foram atribuídos a um centróide, e então recalculando as posições dos pontos K.
4. Repetir as etapas 2 e 3 até que os centroides não se movam mais. Produzindo assim, uma separação dos objetos em grupos, permitindo o cálculo da métrica de minimização.

Usualmente a distância euclidiana é utilizada no algoritmo K-means, entretanto, quando se lida com um vetor de características binárias, a distância euclidiana não é a métrica escolhida, e o vetor médio é indefinido. Logo, uma distância razoável e eficaz entre vetores binários é a distância de Hamming¹, ([GRANA et al., 2013](#)). No entanto, uma abordagem mais específica para ser utilizada concomitantemente ao algoritmo ORB, é a abordagem da técnica conhecida como Brute Force Matcher, que encontra o descritor mais próximo a outro conjunto, utilizando-se da força bruta, ou seja, testando cada um dos outros conjuntos até encontrar a maior similaridade entre eles.

E finalmente, o algoritmo processa uma coleção de vetores binários e procura inicialmente de forma aleatória e recursivamente por um número de bons centroides, que se tornará o dicionário visual para o modelo do BoK (Bag of Keypoints).

¹ Determina o número de bits diferentes nas suas posições correspondentes

Assim sendo, com a finalização da construção do dicionário de keypoints é possível extrair as características de cada símbolo, obtendo um vetor quantizado com as propriedades e características de cada parte da partitura segmentada.

O passo subsequente consiste em treinar um classificador sobre os dados extraídos, e por consequente iniciar a etapa de classificação dos símbolos da partitura. Para tal, foi escolhido o método de reconhecimento de padrões, o SVM – Máquinas de Vetores Suporte, baseado na teoria do aprendizado estatístico.

3.8 SVM - Support Vector Machine

O SVM (Support Vector Machine), deriva-se de uma implementação da máquina de vetores de suporte do pesquisador russo Vladimir N. Vapnik, (CORTES; VAPNIK, 1995) e (VAPNIK; VAPNIK, 1998) em conjunto com outros pesquisadores. A máquina de vetores de suporte tem sido utilizada nos mais diversos campos da computação e análise estatística como: classificação de imagens, reconhecimento de caracteres manuscritos, categorização de texto e hipertexto, bioinformática (classificação de proteínas, detecção de câncer...), reconhecimento de faces entre outras inúmeras aplicações (CAI et al., 2002), (BARZILAY; BRAILOVSKY, 1999).

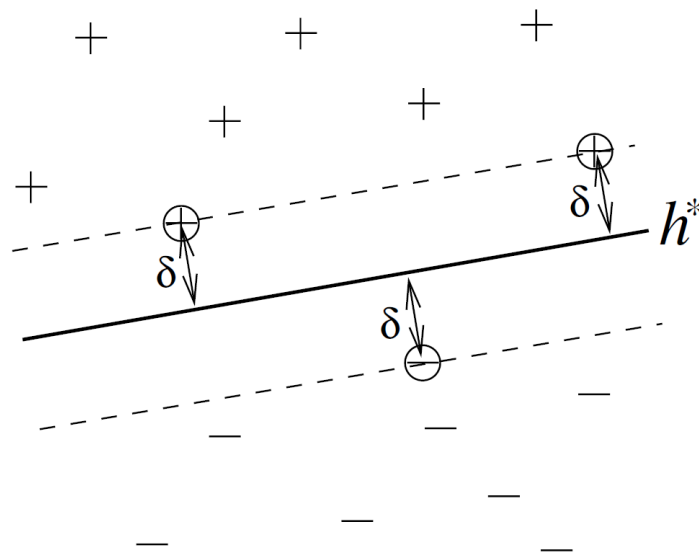
Abordando o problema de modo central, a partir de um conjunto de dados binários, ou seja, o agrupamento possui apenas duas categorias, o algoritmo SVM apresenta um método de otimização para a separação dos dados utilizando um hiperplano, isto é, encontrar a margem máxima de isolamento entre as classes, portanto, posicionando em um mesmo lado da margem o máximo de objetos da mesma classe, e por consequente efetuando a maximização das regiões de borda do hiperplano, identificando assim a região de máxima separação dos dados. Logo, essa área de separação que é composta por pontos de ambas as classes, compõe por esses mesmos pontos, os vetores de suporte que por sua vez são responsáveis pela produção do hiperplano.

No conceito básico de desenvolvimento do SVM, a busca está sempre baseada na separação das classes a partir de uma função linear. Circunstância que raramente pode ser observada quando se trabalha com dados reais. Pois, dificilmente encontramos agrupamentos de objetos que podem ser segmentados perfeitamente ou ao menos com uma boa margem de separação. Para tanto, o SVM introduz o espaço de características orientado por kernel (núcleo), que traduz por sua vez, o espaço de entrada dos dados em um de maior dimensão, onde os objetos tornam-se separáveis, usualmente conhecida como o truque do kernel. Como foi verificado, o SVM foi projetado primordialmente para realizar análise binária, ou seja, classificação de duas classes. Muito embora, algumas extensões tenham sido propostas, como as decomposições um-contra-todos e um-contra-um, para a utilização multiclasse do SVM (MAYORAZ; ALPAYDIN, 1999), o que habilita o seu

uso para aplicações que contenham mais de duas classes (HSU; LIN, 2002) e (SUYKENS; VANDEWALLE, 1999).

Na figura 11, temos o Conceito da máquina de vetores de suporte, para utilização em uma classificação binária, com o hiperplano representado pelo h^* , separando com a máxima margem δ possível os objetos negativos e positivos. Os objetos circulosados representam os vetores de suporte.

Figura 11 – Ilustração de um hiperplano ótimo para dois conjuntos de dados



Fonte – (BEN-NAIM; FRAUENFELDER; TOROCZKAI, 2004)

A função de aprendizado das regras decisão linear do SVM, situa-se na forma,

$$h(x) = \text{sgn}(w * x + b), \quad (3.21)$$

Que são descritos como um vetor de pesos (w), um limiar b , e as propriedades de entrada x . Derivados a partir de uma amostra de entrada n , sendo identificados como, $S_n = ((x_1, y_1) \dots (x_n, y_n))$, como no exemplo citado, temos então duas classes, que podem ser definidas em $S_1 = (y_1 = -1)$ e $S_2 = (y_2 = +1)$, onde, $x_i \in \mathbb{R}^n$ e $y_i \in \{-1, +1\}$. Logo, para duas classes linearmente separáveis, o SVM é capaz de encontrar por meio da máxima distância euclidiana o hiperplano das amostras. Sendo a distância representada pela margem máxima δ , como pode ser visto na Figura 11.

Pois segundo o teorema baseado no princípio de Minimização do Risco Estrutural (SRM), proposta também por (VAPNIK, 1992), que pressupõe a limitação associada a dois fatores: a taxa de erros na etapa de testes sobre a retorno da acurácia para os erros de treinamento e pelo termo da dimensão de Vapnik-Chervonenkis ou dimensão VC (é uma

medida, definida como a quantidade do maior grupo de pontos, nos quais o algoritmo de classificação pode segmentar os dados em conjunto) (VAPNIK; CHERVONENKIS, 2015), sendo o último responsável pelo encargo central na teoria da aprendizagem estatística (HANUSIAK et al., 2010).

Concluindo-se então, que a Minimização do Risco Estrutural (SRM) garante um único hiperplano ótimo possuindo a maior margem sobre os vetores de suporte (CHAVES, 2006).

Ademais, temos a adoção de “variáveis de folga”, ξ , para a utilização em conjuntos de treinamento não-lineares. Pois permitem o equilíbrio entre o risco estrutural e o risco empírico (o valor da taxa da qualidade do ajuste dos dados de treinamento).

Em situações em que os dados não são linearmente separáveis, poderíamos resolver o problema, por exemplo, utilizando funções polinomiais ou círculos, muito embora, esse tipo de abordagem invalide a proposta da utilização dos hiperplanos, que além de apresentar-se como mais custoso, tanto no quesito processamento, quanto por consequente, o tempo para o refinamento de uma curva específica para o conjunto de dados apresentados.

Portanto, fez-se necessário a busca de uma solução computacionalmente eficiente. Encontrada na utilização dos métodos de funções kernels. As funções do kernel, permitem a realização de operações em um espaço de recursos de alta dimensão. Sem, necessariamente, ter de calcular as coordenadas dos dados nesse espaço, mas, simplesmente retornar o produto escalar das imagens dos dados do espaço dos objetos.

Tabela 2 – A tabela apresenta os principais kernels utilizados em aplicações com máquinas de vetores de suporte.

Tipo do Kernel	Expressão	Comentário
Linear	$K(x_i, x_j) = x_i^T * x_j$	Classes linearmente separáveis.
Polinomial - Grau p	$K(x_i, x_j) = (x_i^T * x_j + 1)^p$	O valor da potência de p, deve ser estimado.
RBF - Gaussiano	$K(x_i, x_j) = \exp(-\gamma \ x_i - x_j\ ^2), \gamma > 0$	O valor da amplitude, γ , deve ser estimado.

Na Tabela 2, encontramos os principais kernels utilizados na abordagem das máquinas de vetores de suporte, temos que, $K(x_i, x_j) \equiv \phi(x_i^T) \phi(x_j)$, chamada de função kernel.

Logo, o cálculo da posição do hiperplano, depende da minimização do problema

quadrático convexo, Equação 3.22.

$$V(w, b, \xi) = \frac{1}{2}w * w + C \sum_{i=1}^n \xi_i \quad (3.22)$$

A margem δ , resultante do hiperplano encontrado, corresponde a:

$$\delta = \frac{1}{\|w\|} \quad (3.23)$$

Sendo, que o fator C, determina a relação da negociação feita entre a taxa de erro de treinamento versus a complexidade do modelo criado.

Utilizando-se do mecanismo da mudança de dimensão, foi possível ao SVM ser generalizado para realizar classificações sobre dados não-lineares, tornando-os linearmente separáveis, permitindo, a máquina de vetores de suporte a capacidade de identificar objetos utilizando espaços complexos, por meio da aplicação de funções kernel retornando um hiperplano que possui as suas margens maximizadas para afinal, segmentar o conjunto de dados propostos em classes bem definidas.

3.9 Métricas de Avaliação

No campo do aprendizado de máquina, existem diversas métricas de avaliação de desempenho que podemos utilizar para verificar a qualidade do modelo produzido. Logo, para estimar a precisão do modelo proposto, faz-se necessário a aplicação de métodos numéricos e estatísticos para a sua avaliação.

Logo abaixo, estão descritas algumas das métricas regularmente utilizadas na área de análise de documentos (FLACH, 2003), que também foram utilizadas nesse estudo.

A acurácia (Accuracy) é a medida de desempenho que podemos considerar mais axiomática, pois trata-se da proporção de objetos corretamente classificados, ou seja, é a relação entre o número de acertos pelo número total de instância disponíveis.

$$Acc = \frac{vp + vn}{vp + fp + vn + fn} \quad (3.24)$$

A precisão (Precision) analisa a proporção de observações que sejam positivas e estejam corretas de uma determinada classe, ou seja, a relação entre os objetos que são da classe e foram identificados corretamente pela adição entre esse mesmo número e os outros objetos incorretamente identificados como pertencentes à essa classe também, mas que pertencem a outra classe (falso positivo).

$$Prc = \frac{vp}{vp + fp} \quad (3.25)$$

O Recall é a proporção de eventos positivos corretamente previstos, ou seja, a relação entre os objetos classificados corretamente como pertencentes a essa classe, pela quantidade total de objetos que pertencem a classe.

$$Rcl = \frac{vp}{vp + fn} \quad (3.26)$$

O F-measure, corresponde à média harmônica entre a precisão e a cobertura (recall), considerando tanto as instâncias falso-positivas quanto as falso-negativas. Métrica também reconhecida por tratar melhor os conjuntos de dados desbalanceados.

$$F1 = \frac{2 * precision * recall}{precision + recall} \quad (3.27)$$

Onde podemos ler, VP como verdadeiro positivo, VN como verdadeiro negativo, FP como falso positivo e FN como falso negativo. Essas são algumas das métricas mais utilizadas na análise de modelos para reconhecimento de partituras manuscritas ([REBELO, 2012](#)), permitindo a observação e análise da técnica proposta sobre outras abordagens igualmente padronizadas.

4 Materiais e Métodos

Com a finalidade de avaliar os métodos e técnicas propostas nesse trabalho, foi utilizado a biblioteca de visão computacional OpenCV¹ na versão 3.1, conjuntamente com o software para mineração de dados Weka², na versão 3.7, ambos open-source, assim como a base de dados CVC-MUSICMA empregada nesse projeto.

A biblioteca de algoritmos de visão computacional OpenCV, originalmente desenvolvida pela Intel no ano 2000 (BRADSKI; KAEHLER, 2008), proporciona aos estudantes e profissionais um conjunto de ferramentas otimizadas para o desenvolvimento de aplicações para reconhecimento de imagens bem como para o aprendizado de máquina. Foram utilizados algoritmos de binarização, segmentação utilizando arrays multi-dimensionais, assim como técnicas de extração e reconhecimento de padrões.

O repositório de algoritmos e metodologias para análise e reconhecimento de padrões, Waikato Environment for Knowledge Analysis (WEKA), foi criado pela Universidade Waikato na nova Zelândia em 1993, desde então, têm sido sistematicamente melhorada e atualizada, correspondendo a uma ótima ferramenta para o aprendizado e análise, (HALL et al., 2009).

Foi utilizada em conjunto com o pacote LIBSVM, usado tanto para a estimação dos parâmetros C e γ , necessários quanto ao uso do kernel RBF, utilizando-se da técnica de “Grid Search”, quanto do processo subsequente de reconhecimento dos símbolos musicais segmentados e posterior extração das métricas de desempenho.

4.1 CVC-MUSICMA (Base de Dados)

A base de dados CVC-MUSICMA, (FORNÉS et al., 2012), contém 1000 partituras manuscritas produzidas por 50 músicos diferentes e, experientes, que reproduziram igualmente em sua própria grafia 20 músicas selecionadas, criada com o intuito de ser uma grande base de dados para a pesquisa na área de identificação de autores e remoção de pautas.

Foram escolhidas 4 partituras de cada músico, para então ser realizado o processo de segmentação manual para extração de 4000 símbolos musicais divididos em 20 classes, que podem ser visualizados na Tabela 3, sendo constituído de 200 símbolos por classe.

¹ <https://www.opencv.org>

² <https://www.cs.waikato.ac.nz/ml/weka>

Tabela 3 – Símbolos musicais utilizados nesse trabalho.











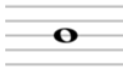









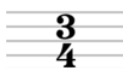



















Digital	Manuscrita	Nome	Classe
		Diminuendo	1
		Barra	2
		Bemol	3
		Semínima	4
		Barra	5
		Semibreve	6
		Clave de Sol	7
		Pausa de Semínima	8
		Mínima	9
		Sustenido	10
		Tempo	11
		Semicolcheia	12
		Pausa de Colcheia	13
		Clave de Dó	14

Tabela 3 – Símbolos musicais utilizados nesse trabalho.

Digital	Manuscrita	Nome	Classe
		Clave de Fá	15
		Crescendo	16
		Cabeça da Nota	17
		Bequadro	18
		Duas Semicolcheias Ligadas	19
		Duas Colcheias Ligadas	20

5 OMR Proposto

O trabalho proposto, implica no reconhecimento de partituras musicais manuscritas, mas como pôde ser constatado nas seções anteriores, o trabalho de aquisição de informações de notas musicais manuscritas, pode ser considerado muito custoso, visto que ainda não existe metodologias robustas para o reconhecimento de notas musicais manuscritas (REBELO, 2012), o que poderíamos considerar o princípio básico para a conversão de uma partitura, problema abordado nesse estudo, além disso há a necessidade da existência de uma base de dados sólida e grande o suficiente para agregar as mais diversas abordagens musicais bem como aspectos culturais que influenciam na estrutura da mesma.

Desse modo, esse trabalho apresenta um tratamento mais específico e detalhado para a segmentação e detecção de notas musicais, assim como uma introdução a sistematização prática do reconhecimento de partituras musicais manuscritas.

Figura 12 – Pipeline representando as seis etapas propostas para o reconhecimento de uma partitura manuscrita.



Fonte – Autor

Definiu-se, conseqüentemente, que o objetivo a ser atingido consiste em extrair o máximo de informações possíveis de uma partitura. Embora o segmento de análise de documentos para partituras digitais, ou seja, aquelas partituras produzidas por meio de sistemas informáticos, esteja bem consolidado com o seu emprego em diversas aplicações

de mercado como: Finale¹, Sibelius², MuseScore³ entre outros.

As partituras manuscritas ainda carecem de metodologias robustas para reconhecimento de grafias diversas, bem como técnicas para tratamento de ruídos provenientes de papéis degradados e ou tinta desgastada, entre muitas outras peculiaridades, como por exemplo, símbolos musicais que estão no verso da partitura, e com o tempo e pelo desgaste da folha, ele começa a ser visto do outro lado também da página, o que influencia diretamente no processo de detecção automático.

Logo, nossa estratégia de reconhecimento está baseada de acordo com a abordagem proposta por Rebelo et al. (2012), Figura 6, na qual um sistema OMR é constituído de várias camadas de análise de informação, que perpassa desde o pré-processamento, processamento das pautas, processamento dos símbolos musicais, classificação, reconstrução da notação musical, e finalmente a publicação do arquivo digital da partitura, como pode ser visto no pipeline da Figura 12.

5.1 Pré-Processamento

A etapa de pré-processamento constitui-se da aquisição de partituras provenientes da base de dados CVC-MUSCIMA, digitalizadas em 300dpi, binarizadas utilizando o método de Otsu, (OTSU, 1979), e realizado a operação morfológica de dilatação, responsável por melhorar a detecção das linhas nessa etapa e conseqüentemente a identificação das pautas.

Essa etapa é capaz de solucionar pequenos ruídos, assim como permitir que segmentos de linhas ganhem mais expressividade e largura, conseqüentemente eliminando pequenas regiões não conectadas das pautas. Como podemos observar na Figura 13.

O processo de dilatação da imagem realizado após a aplicação de um elemento estruturante horizontal, retorna uma imagem quase sem falhas ou interrupções em suas linhas, facilitando o processo de detecção das pautas. Conforme pode ser observado nas Figuras 14 e 15, apresentando respectivamente o antes e depois das operações serem realizadas.

Pois, quando bem aplicada, a dilatação também auxilia na conexão das partes pertencentes de um mesmo símbolo.

¹ <https://www.finalemusic.com>

² <https://www.avid.com/sibelius>

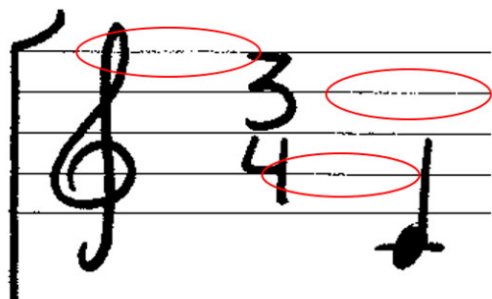
³ <https://www.musescore.com>

Figura 13 – Mostra o antes (a) e depois (b) da imagem da partitura escolhida, na qual foi empregada o método de binarização de Otsu, e a operação morfológica de dilatação.



Fonte – (a) - (FORNÉS et al., 2012); (b) - Autor

Figura 14 – A imagem apresenta em vermelho as falhas, os gaps comumente encontrados nas linhas de pauta manuscrita.



Fonte – Autor

Figura 15 – Apresenta a mesma imagem após ser efetuada a dilatação.



Fonte – Autor

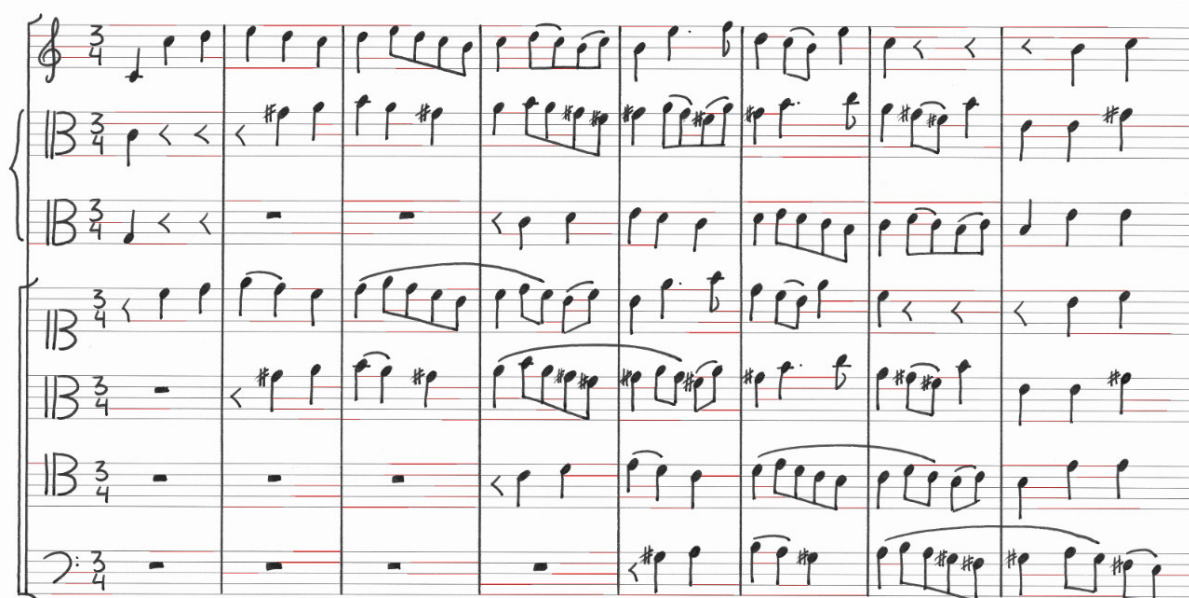
5.2 Processamento das pautas

As pautas por sua vez, são uma importante estrutura da notação musical, pois são elas que informam qual a posição da clave e por conseqüente todas as posições de todos os outros símbolos, assim como, também interferem no reconhecimento das notas e símbolos por estar em sobreposição a eles, sendo necessário, portanto, a sua identificação e remoção.

As linhas de pauta, foram detectadas utilizando basicamente o algoritmo de Dalitz, com algumas modificações, como por exemplo, a bem reconhecida transformada de Hough (ILLINGWORTH; KITTLER, 1988), método muito utilizado na detecção de formas

parametrizadas, como linhas, círculos, elipses, etc. Empregada na detecção de segmentos de pauta, como pode ser visualizado na Figura 16.

Figura 16 – Os segmentos de pauta detectados estão em vermelho.

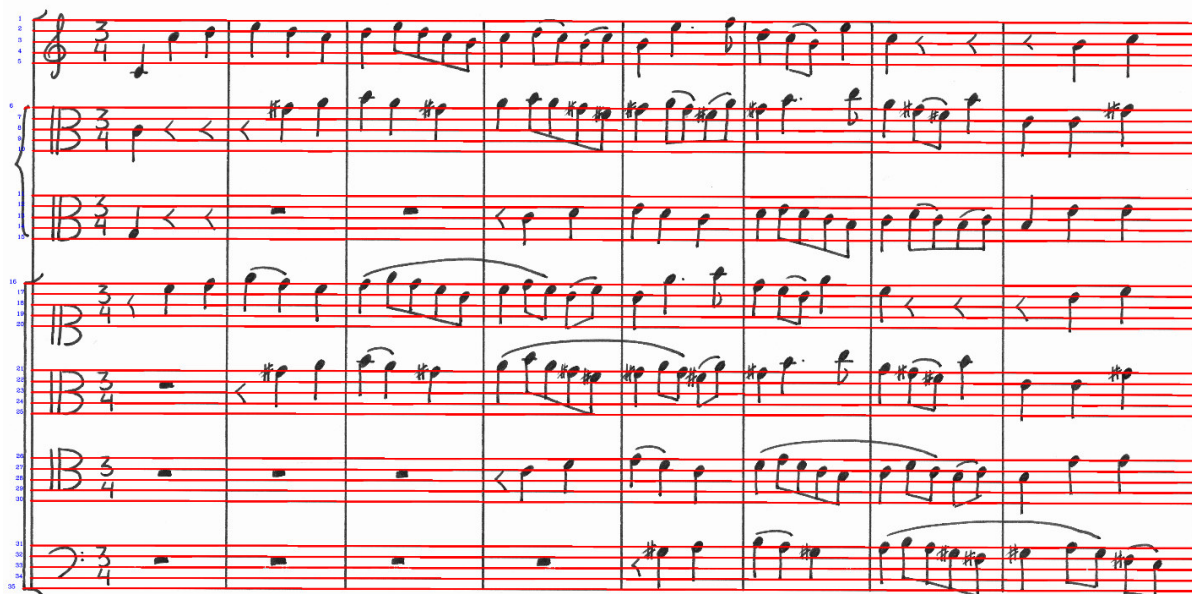
A musical score for a piano piece in 3/4 time, featuring a treble and two bass staves. The score is divided into eight measures. Red brackets are drawn under various notes and rests across all staves, indicating detected segments. The notes are mostly eighth and sixteenth notes, with some accidentals (sharps and naturals). The rests are represented by horizontal lines on the staves.

Fonte – Autor

Os segmentos detectados são então empregados na formação de um grafo com caminhos de menor custo. Os segmentos são ordenados por suas posições em altura e largura, então é verificado o caminho estável entre as duas margens utilizando o algoritmo, proposto por Dalitz chamado de Line Track Height (LTH) (DALITZ; KARSTEN; POSE, 2005), que verifica se caminho percorrido é duas vezes menor que a altura da linha de pauta, pesquisa pixel a pixel com a finalidade de conectar os segmentos detectados que estejam sobre a linha ou mais próximo possível a ela, enquanto que os segmentos não pertencentes aquela linha de pauta, identificados por meio da utilização de variáveis de folgas são removidos. As variáveis de folga são utilizadas nesse momento para permitir ao algoritmo o retorno com mais precisão sobre a posição de um segmento de linha, que poderia ser descartado por não estar necessariamente conectada aos outros pixels da linha. Garantindo uma boa detecção, mesmo sobre linhas de pautas curvadas e inclinadas e verificando com mais precisão a posição dos símbolos para cada posição nas linhas de pauta. O exemplo contido na Figura 17, nos mostra o resultado da metodologia aplicada.

Portanto, após a aplicação do método, foi possível identificar a cardinalidade e respectivas posições das pautas da partitura e as suas respectivas linhas. Também é possível definir uma regra pertencente a notação musical, como a verificação dos múltiplos de cinco para o número de linhas das pautas. Pois caso o número de linhas de pauta, sofra divergências quanto aos múltiplos de cinco, provavelmente o processo de detecção não

Figura 17 – Detecção das linhas de pauta em vermelho.



Fonte – Autor

foi preciso o suficiente, subtraindo o número correto de linhas, ou adicionando linhas de pauta, que não estão presentes.

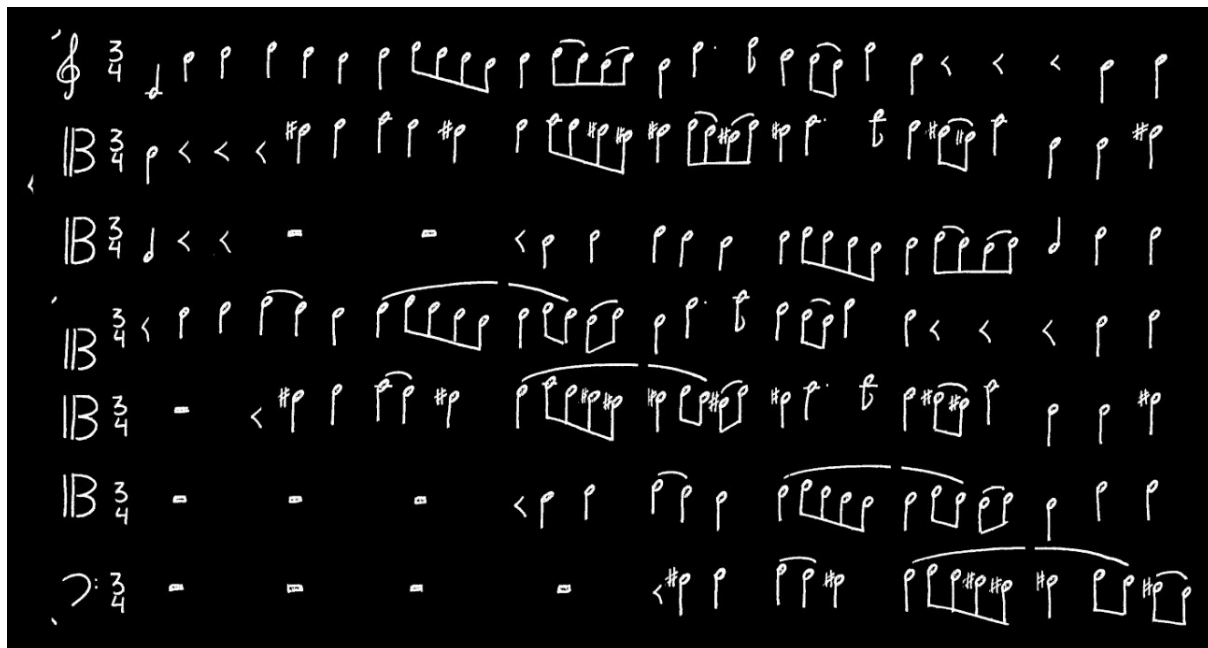
5.3 Processamento dos símbolos musicais

A próxima etapa, consiste na remoção das linhas detectadas, embora alguns pesquisadores tenham efetuado a etapa subsequente, utilizando símbolos que possuíam partes da pauta, o trabalho apresentado optou pela remoção das mesmas, utilizando as operações morfológicas de erosão e dilatação. O elemento estruturante construído, se valeu das informações da etapa anterior para aquisição dos dados acerca do tamanho das pautas, permitindo a sua escalabilidade independente da partitura utilizada, vejamos a seguir o resultado do processo de remoção.

O processo seguinte, consiste na localização e segmentação automática dos símbolos. A ideia aqui, reside em encontrar as regiões conectadas, como possuímos uma imagem binarizada, possuindo apenas 0's (fundo preto) ou 255's (símbolos - branco), precisaremos apenas de um algoritmo que percorra a nossa matriz em busca de uma célula branca, e a seguir encontre todos os vizinhos imediatamente próximos, que também apresentem o valor 255.

Utilizando o algoritmo FloodFill, que é capaz de retornar um array contendo todas as regiões conectadas entre si, pela mesma cor. Foi possível realizar a demarcação das notas

Figura 18 – Partitura após etapa de remoção das pautas.



Fonte – Autor

e símbolos por meio de uma bound-box, e então executando a sua extração da partitura e posterior classificação. Um exemplo do método relatado, pode ser observado na Figura 16.

Os símbolos segmentados, são submetidos a etapa de classificação. Para tanto, se faz necessário a execução da etapa de treinamento do classificador utilizado, logo foram selecionados manualmente um conjunto de 20 símbolos musicais, Tabela 3 a serem treinados e validados em um processo de cinco etapas para a extração e detecção das características, Figura , para compor as amostras de treinamento. Gerando um arquivo que possui 4000 imagens, contendo 400 imagens por classe.

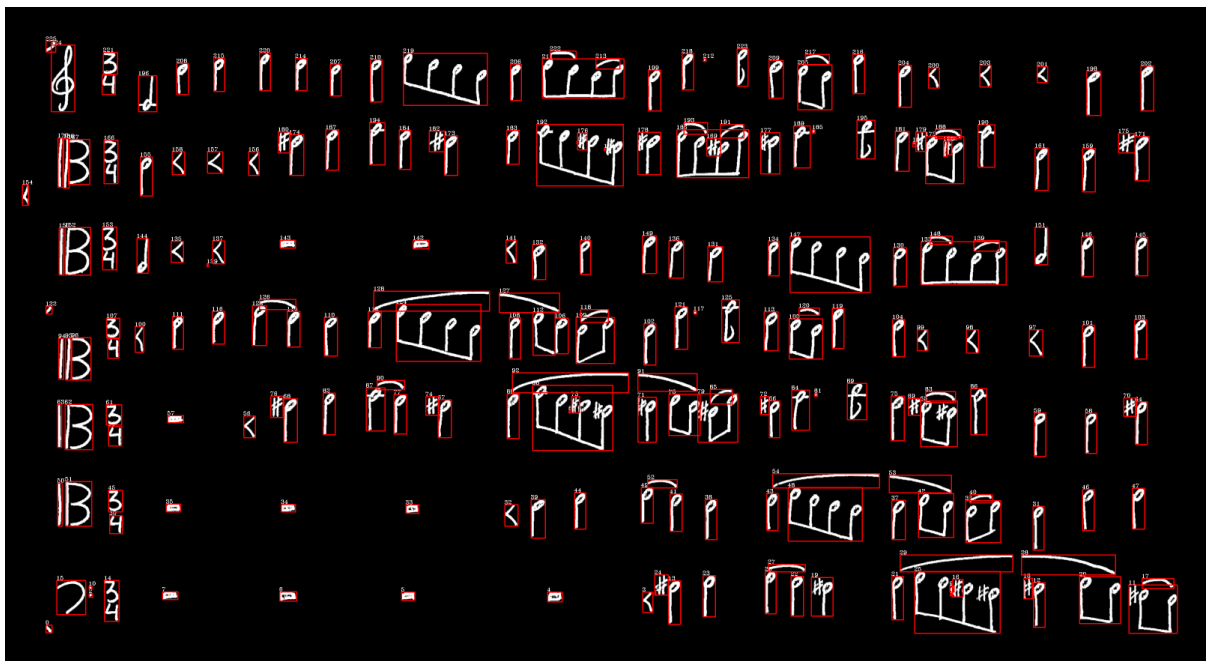
5.3.1 Extração de Características - Símbolos Segmentados

É importante ressaltar que algumas das imagens selecionadas, contêm alguns ruídos, como partes de outros símbolos ou erros de grafia, não sendo aplicados filtros de correção, como o filtro gaussiano, ou manipulações manuais sobre os símbolos, pois pretende-se atestar a automatização do processo mesmo sobre condições não favoráveis.

Pois conforme pode ser visto na Figura 16, as imagens das partituras não estão danificadas pelo tempo, mal-uso ou mesmo armazenamento incorreto, o que difere muito, das propriedades de uma partitura manuscrita usualmente encontrada.

Portanto, o fluxo da nossa metodologia escolhida no projeto para a classificação dos símbolos manuscritos, corresponde ao pipeline ilustrado na Figura 20.

Figura 19 – Símbolos da partitura segmentados em vermelho.



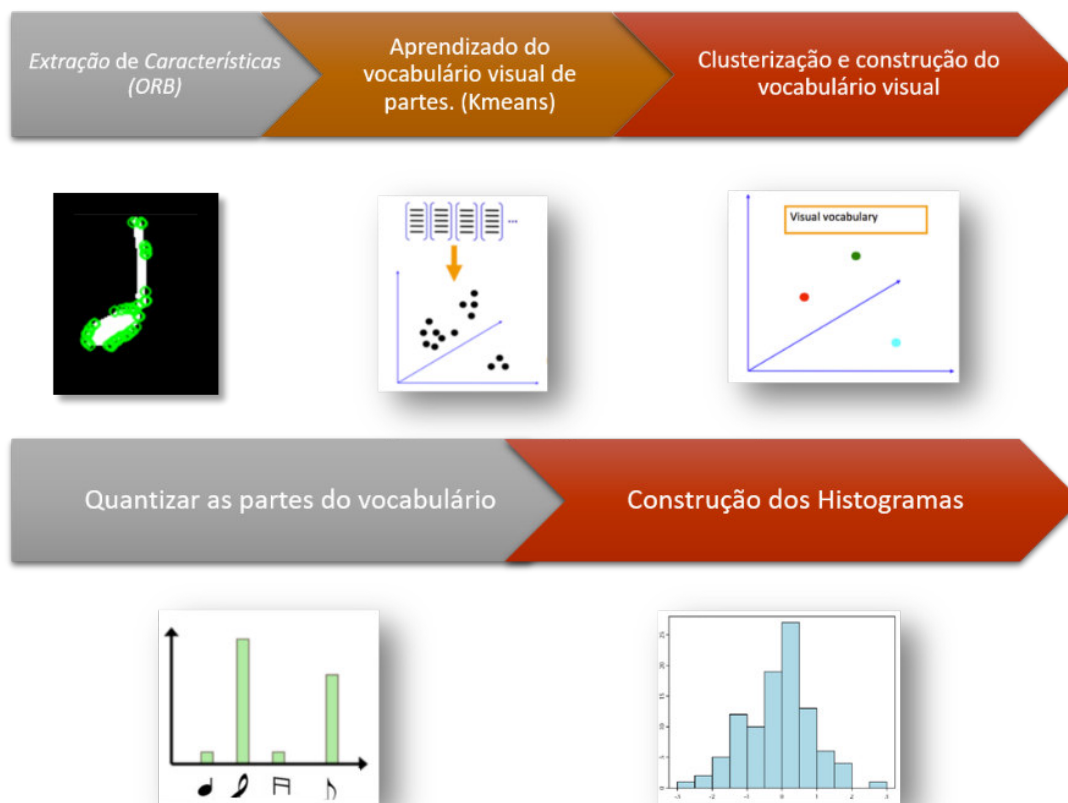
Fonte – Autor

As imagens segmentadas e binarizadas são então adicionadas e processadas sobre o método ORB, utilizando-se como matcher o algoritmo de Brute Force (esta técnica procura em todos os vetores de características, quais vetores mais se aproximam, utilizando uma métrica de distância, conhecido como Brute Force Matcher, sobre a distância de Hamming (utilizada para denotar a diferença entre duas cadeias binárias). Como utilizou-se o método holdout, para garantir a generalização dos dados do nosso modelo, a construção do dicionário foi repetida inúmeras vezes, para assegurar um bias baixo para os dados, retornando 577 características que foram armazenadas em arquivo, no formato xml. Para cada instância de teste e treinamento, descritos detalhadamente na seção dos resultados, foram utilizados rigorosamente conjuntos de dados mutuamente exclusivos, tornando-se necessário a reconstrução do vocabulário visual, bem como a clusterização dos dados a cada nova etapa de treinamentos e testes, e finalmente retornar o arquivo contendo os histogramas de características para o classificador SVM.

5.4 Classificação

A etapa de classificação, consiste em utilizar os histogramas de características, como os vetores de entrada no classificador SVM. Sendo que, a metodologia empregada consistiu em treinar e testar as amostras utilizando-se a técnica holdout, proporcionando a verificação do modelo quanto a sua capacidade de generalização.

Figura 20 – Pipeline para a detecção dos símbolos musicais manuscritos utilizando o ORB. Os símbolos foram extraídos na etapa de pré-processamento.



Fonte – Autor

A validação cruzada, representa a busca pela definição da precisão empírica do modelo com os dados reais, ou seja, como a arquitetura de classificação vai se comportar com outro conjunto de dados não conhecidos, o que representa testar o modelo no “mundo real”. A concepção da validação está baseada essencialmente na separação dos dados em dois subconjuntos mutuamente exclusivos. Os quais são escolhidos posteriormente, um para servir de amostrar para o treinamento e conseqüentemente estimação dos parâmetros e o outro como os dados reais para teste e validação (KOHAVI et al., 1995). Muitas metodologias de particionamento foram propostas, as três mais conhecidas e utilizadas são: o k-fold, holdout e o leave-one-out.

O método holdout, às vezes chamado de teste de estimativa de amostras, aplicado nesse trabalho, apresenta a estrutura de particionamento em dois subconjuntos, como citado, os de teste e os de treinamento. Portanto, as amostras foram divididas em 4 sub-grupos mutualmente exclusivos, sendo repetidos por cinco vezes a etapa de classificação e validação. Os dados foram particionados em grupos de: treino: 20% e teste: 80%, treino:

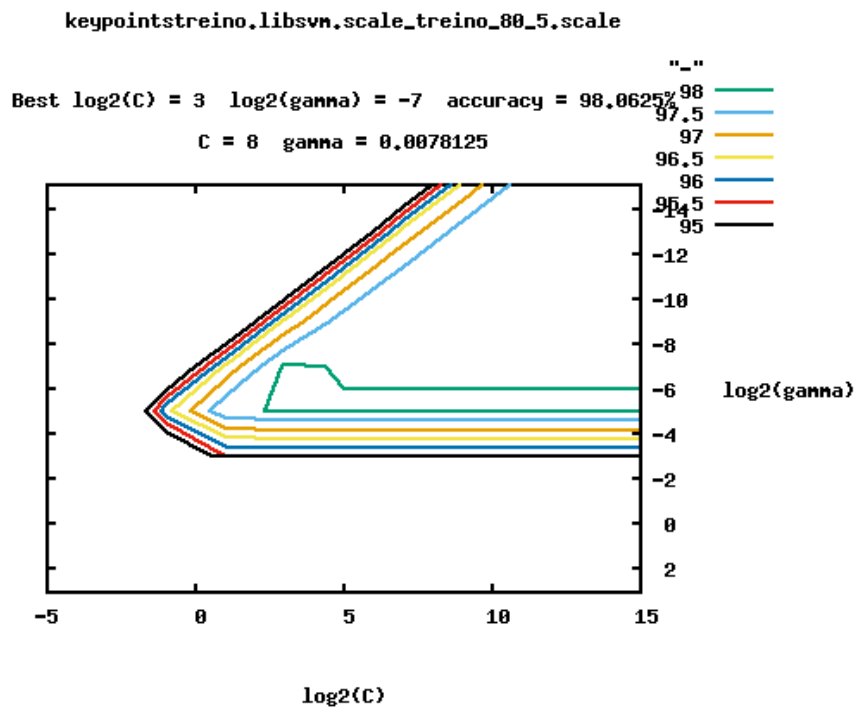
40% e teste: 60%, treino: 60% e teste: 40% e treino: 80% e teste: 20%.

Também precisamos estimar os parâmetros C e γ utilizados do núcleo radial, logo, uma pesquisa de seleção de modelo precisa ser efetuada, para que o classificador possa prever com precisão os dados submetidos posteriormente, isto é, dados desconhecidos. A solução encontrada está na utilização da pesquisa em grade ou a normalmente conhecida Grid-Search sobre os parâmetros C e γ .

No processo de execução do Grid-Search vários testes são realizados sobre o par (C, γ) , e ao final aquele que possuir a melhor precisão é o escolhido. As tentativas de valores sequenciais crescentes e exponenciais, demonstrou-se um método prático para a identificação de bons parâmetros (HSU et al., 2003). Exemplos de especificações para C e γ , temos $C = 2^{-8}, 2^{-5}, \dots, 2^{16}$; $\gamma = 2^{-16}, 2^{-12}, \dots, 2^2$.

Na Figura 21, é possível visualizar o comportamento da busca, utilizando o Grid-Search.

Figura 21 – Grid-Search - Parâmetros $C = 8$ e $\gamma=0.0078125$, com uma acurácia de 98.0625%.



Fonte – Autor

Logo, é possível concluir que a sistematização do reconhecimento de símbolos e sua validação apresentam resultados compatíveis e confiáveis, permitindo a realimentação do sistema, a partir de novas instâncias detectadas, propiciando um ciclo de construção robusto e adaptável.

5.4.1 Resultados

Os resultados obtidos, para a classificação de 4000 símbolos, divididos em 20 classes, estão categorizados em 04 instâncias de treinamentos e testes, na Tabela 5, construídas conforme especificado na seção classificação e validação. Também é possível visualizar os parâmetros estimados do kernel RBF. A acurácia detalhada na Tabela 5, representa a precisão geral para todas as 400 instâncias de cada uma das classes propostas utilizando-se a validação cruzada.

Tabela 4 – Apresenta os dados para a etapa de estimação dos parâmetros do kernel RBF, bem como os resultados da classificação dos símbolos musicais manuscritos, para cada etapa da análise da validação cruzada.

Classificação Treino / Teste	Resultados	Acurácia	Cardinalidade - VP, Teste
20/80 - 1	$C = 8.0, \gamma = 0.03125, \text{rate}=95.75$	94,0625%	(3010/3200)
20/80 - 2	$C = 32.0, \gamma = 0.0078125, \text{rate}=94.5$	95,2813%	(3049/3200)
20/80 - 3	$C = 2.0, \gamma = 0.03125, \text{rate}=95.875$	94,4688%	(3023/3200)
20/80 - 4	$C = 8.0, \gamma = 0.0078125, \text{rate}=95.125$	94,6563%	(3029/3200)
20/80 - 5	$C = 8.0, \gamma = 0.0078125, \text{rate}=95.125$	96,0313%	(3073/3200)
Média Percentual		94,90%	
40/60 - 1	$C = 8.0, \gamma = 0.03125, \text{rate}=96.8125$	97%	(2328/2400)
40/60 - 2	$C = 8.0, \gamma = 0.03125, \text{rate}=97.625$	96,5833%	(2318/2400)
40/60 - 3	$C = 8.0, \gamma = 0.03125, \text{rate}=97.8125$	96,75%	(2322/2400)
40/60 - 4	$C = 2.0, \gamma = 0.03125, \text{rate}=97.5$	96,5417%	(2317/2400)
40/60 - 5	$C = 8.0, \gamma = 0.03125, \text{rate}=97.3125$	96,75%	(2322/2400)
Média Percentual		96,7250%	
60/40 - 1	$C = 8.0, \gamma = 0.03125, \text{rate}=97.7083$	97,5%	(1560/1600)
60/40 - 2	$C = 8.0, \gamma = 0.03125, \text{rate}=97.3333$	98,3125%	(1573/1600)
60/40 - 3	$C = 2.0, \gamma = 0.03125, \text{rate}=97.25$	97,875%	(1566/1600)
60/40 - 4	$C = 8.0, \gamma = 0.03125, \text{rate}=97.75$	97,25%	(1556/1600)
60/40 - 5	$C = 2.0, \gamma = 0.03125, \text{rate}=97.8333$	97,9375%	(1567/1600)
Média Percentual		97,775%	
80/20 - 1	$C = 8.0, \gamma = 0.03125, \text{rate}=97.875$	98,125%	(785/800)
80/20 - 2	$C = 8.0, \gamma = 0.03125, \text{rate}=97.875$	98,5%	(788/800)
80/20 - 3	$C = 2.0, \gamma = 0.03125, \text{rate}=97.5625$	98,625%	(789/800)
80/20 - 4	$C = 8.0, \gamma = 0.03125, \text{rate}=98.25$	97,375%	(779/800)
80/20 - 5	$C = 8.0, \gamma = 0.0078125, \text{rate}=98.0625$	97,625%	(781/800)
Média Percentual		98,050%	

Como é possível ser analisado, os melhores resultados foram obtidos no caso 80/20 com uma pontuação média total de acertos de 98.05%. É interessante notar, como a acurácia vai melhorando conforme aumentam o número de objetos que constituem a etapa de treinamento do SVM. Também vale notar, que a variação na taxa percentual de acerto para cada uma das 4 etapas de validação, não sofrem grandes mudanças, como por exemplo na etapa 60/40 o percentual de acerto sempre ficou em aproximadamente 97%.

O que nos leva a igualmente considerar os valores encontrados para os parâmetros

C e γ , que apresentam sempre valor próximos uns dos outros dentro da mesma etapa, o que nos permite atestar a capacidade de generalização do nosso modelo, mas sem incorrer no problema do super ajustamento ou overfitting. Vejamos agora em detalhes, as métricas da melhor abordagem de classificação para o grupo 80/20, especificamente o 3º ciclo, com uma acurácia de 98,62%. Apresentados na Tabela 5.

Tabela 5 – Apresenta os dados para a etapa de estimação dos parâmetros do kernel RBF, bem como os resultados da classificação dos símbolos musicais manuscritos, para cada etapa da análise da validação cruzada.

Classe	VP - %	FP - %	Precisão	Recall	F-Measure	MCC	Área ROC	Área PRC
1	1	0	1	1	1	1	1	1
2	1	0	1	1	1	1	1	1
3	0,95	0,001	0,974	0,95	0,962	0,96	0,974	0,928
4	0,975	0	1	0,975	0,987	0,987	0,988	0,976
5	0,975	0	1	0,975	0,987	0,987	0,988	0,976
6	1	0	1	1	1	1	1	1
7	1	0	1	1	1	1	1	1
8	1	0,001	0,976	1	0,988	0,987	0,999	0,976
9	1	0	1	1	1	1	1	1
10	1	0	1	1	1	1	1	1
11	1	0,003	0,952	1	0,976	0,975	0,999	0,952
12	0,975	0	1	0,975	0,987	0,987	0,988	0,976
13	0,975	0,001	0,975	0,975	0,975	0,984	0,987	0,952
14	1	0	1	1	1	1	1	1
15	1	0	1	1	1	1	1	1
16	0,975	0	1	0,975	0,987	0,987	0,988	0,976
17	1	0	1	1	1	1	1	1
18	1	0	1	1	1	1	1	1
19	0,975	0,007	0,886	0,975	0,929	0,926	0,984	0,865
20	0,925	0,001	0,974	0,925	0,949	0,946	0,962	0,904

O desempenho apresentado para a classificação de símbolos musicais manuscritos, utilizando 3200 amostras para treinamento, demonstrou-se eficaz e pronto a ser empregado. Além disso, convém ressaltar a expectativa de aperfeiçoamento na adição de novos conjuntos de dados. Ademais, outra métrica muito utilizada para análise, consiste no aproveitamento dos dados presentes na matriz de confusão. Na matriz de confusão é possível identificar, por exemplo, quais são as classes com as maiores taxas de falso positivos, e por consequente, identificar qual a classe responsável pelo aumento das taxas de erros. Observando a matriz de confusão para uma taxa erros próximo a 1,37% na Figura 22.

Analisemos então, as características estruturais das classes de símbolos, que apresentaram inconsistências com o modelo proposto. Vale destacar, que a principal característica consiste no formato do símbolo, pois questões como luminância, textura, cor, entre outras, não são consideradas sobre imagens binarizadas. Portanto, na Tabela 6, relacionamos alguns dos erros mais comuns no processo de classificação do nosso modelo.

Os símbolos apresentados na Tabela 6, retratam as características estruturais que compõe as notas musicais responsáveis pelas taxas de falso positivos. Pois como é possível

Figura 22 – Exemplo da matriz de confusão para o melhor resultado encontrado.

```

=== Confusion Matrix ===
  a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p  q  r  s  t  <-- classified as
40  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | a = 0
 0 40  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | b = 1
 0  0 38  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0 | c = 2
 0  0  1 39  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | d = 3
 0  0  0  0 39  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0 | e = 4
 0  0  0  0  0 40  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | f = 5
 0  0  0  0  0  0 40  0  0  0  0  0  0  0  0  0  0  0  0  0 | g = 6
 0  0  0  0  0  0  0 40  0  0  0  0  0  0  0  0  0  0  0  0 | h = 7
 0  0  0  0  0  0  0  0 40  0  0  0  0  0  0  0  0  0  0  0 | i = 8
 0  0  0  0  0  0  0  0  0 40  0  0  0  0  0  0  0  0  0  0 | j = 9
 0  0  0  0  0  0  0  0  0  0 40  0  0  0  0  0  0  0  0  0 | k = 10
 0  0  0  0  0  0  0  0  0  0  0 39  0  0  0  0  0  0  1  0 | l = 11
 0  0  0  0  0  0  0  1  0  0  0  0 39  0  0  0  0  0  0  0 | m = 12
 0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0  0  0  0  0 | n = 13
 0  0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0  0  0  0 | o = 14
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 39  0  0  1  0 | p = 15
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0  0 | q = 16
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0 | r = 17
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 39  1 | s = 18
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  3 37 | t = 19
    
```

Fonte – Autor

observar, algumas regiões da forma de um símbolo podem ser encontradas facilmente na outra nota musical classificada incorretamente.

É importante notar que o algoritmo utilizado na extração de características, influencia diretamente nesse resultado, pois sendo invariante a rotação, os pontos de interesse extraídos denotam essas propriedades, como pode ser visto na classificação imprecisa entre as classes 9 e 3, a nota mínima e o bemol, respectivamente.

As quais possuem estruturas gráficas idênticas, apenas diferindo por meio de um espelhamento horizontal. Portanto, ficou claro, que mesmo diante dos benefícios da utilização de algoritmos invariantes a escala e rotação, encontramos alguns obstáculos quando temos símbolos que necessitem de informações sobre o seu estado de rotação. Ou mesmo quanto a escala, pois notadamente a nota musical bemol possui a sua forma padrão menor que a do símbolo musical da nota mínima.

Tabela 6 – Apresenta alguns dos erros mais comuns de classificação, identificados por suas classes e respectiva imagem. VP: Verdadeiro Positivo, FP: Falso Positivo.













Classe - VP	Símbolo - VP	Classe - FP	Símbolo - FP
3		11	

Tabela 6 – Apresenta alguns dos erros mais comuns de classificação, identificados por suas classes e respectiva imagem. VP: Verdadeiro Positivo, FP: Falso Positivo.

Classe - VP	Símbolo - VP	Classe - FP	Símbolo - FP
5		13	
9		3	
12		19	
13		8	
19		20	

Embora, existam alguns problemas com no uso do ORB, a sua metodologia de extração de características ainda apresenta ótimos resultados em comparação com outras abordagens realizadas sobre a mesma base de dados, como pode ser analisado na Tabela 7, pois a sistematização apresentada utilizando o ORB + BoK (Bag of KeyPoints) + SVM, forneceu resultados superiores a abordagens como a utilização dos momentos de HU, a qual consiste no método de fornecer características relativas a sete momentos invariantes a escala, rotação e translação (HU, 1962). Assim como exemplificado na técnica de aprendizado de máquina, as redes neurais convolucionais com a GoogLeNet, a rede foi aplicada sobre a mesma base de dados, obtendo resultados próximos a 97,3%.

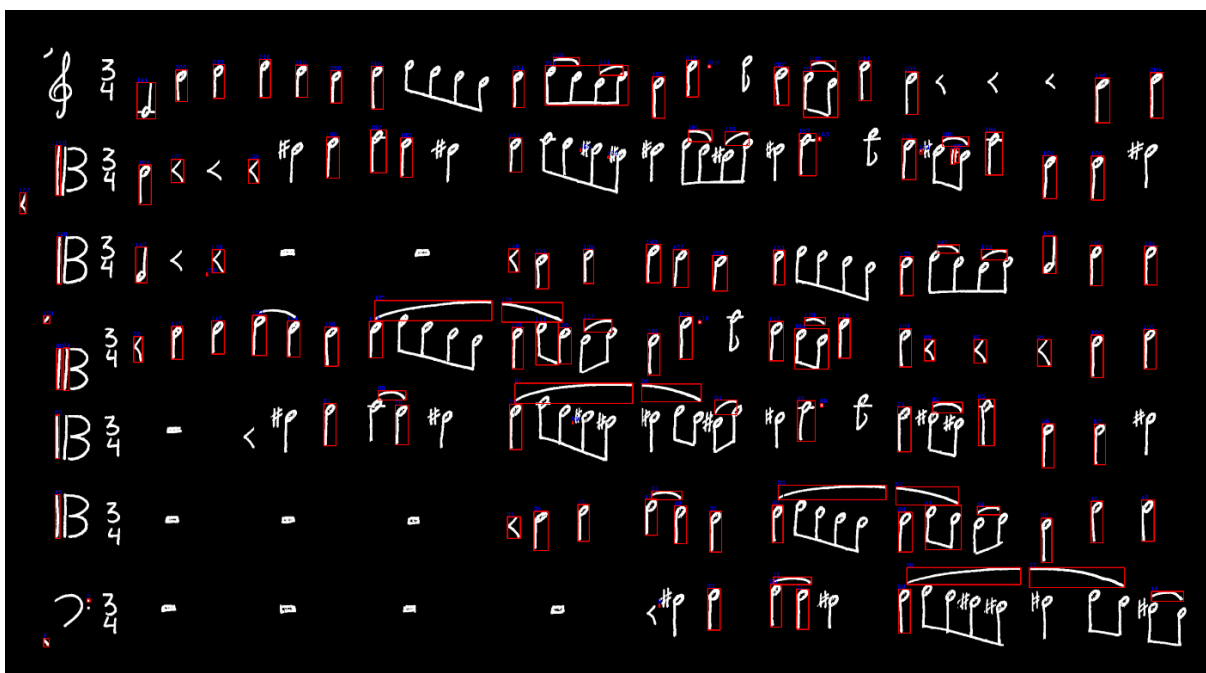
Tabela 7 – Observa-se os resultados de três abordagens diferenciadas sobre o problema proposto neste artigo com a mesma base de dados, mesmos símbolos e método holdout.

Métodos Implementados	Acurácia Total
Momentos de HU + SVM	71,12%
Redes Neurais Convolucionais - GoogLeNet (PEREIRA et al., 2016)	97,21%
ORB + BoF + SVM	98,05%

Na Figura 23, temos um exemplo prático sobre a aplicação da metodologia sobre uma partitura manuscrita. Observe, o resultado da classificação para a nota semínima.

É possível perceber, que de modo geral, a classificação para a nota semínima atingiu um nível de acurácia notável para o percentual das amostras verdadeiramente positivas, embora ainda tenha ocorrido taxas consideráveis de classificação para falso negativo, falso positivo e verdadeiro negativo, como pode ser observado nas notas com ligadura, bem

Figura 23 – A figura apresenta em vermelho os símbolos da partitura classificados como semínima



Fonte – Autor

como com legato (o símbolo da linha curvada que é empregado por cima ou por baixo das notas, permitindo ligá-las entre si), assim como para a clave de dó, que possui uma barra não ligada ao corpo principal do símbolo, permitindo que o algoritmo FloodFill segmente a nota que deveria ser considerada como um todo, transformando-a em dois símbolos, enquanto na verdade haveria somente um.

Um problema que pode ser solucionado, quando empregado o uso das regras semânticas da notação musical para a detecção de claves, valendo-se da sua posição em relação a pauta.

Portanto, os resultados acima demonstram a capacidade da metodologia de lidar com o reconhecimento de símbolos musicais manuscritos pertencentes a diferentes autores, bem como diferentes formas de grafias, resultando em uma metodologia eficiente para a classificação de notas musicais manuscritas.

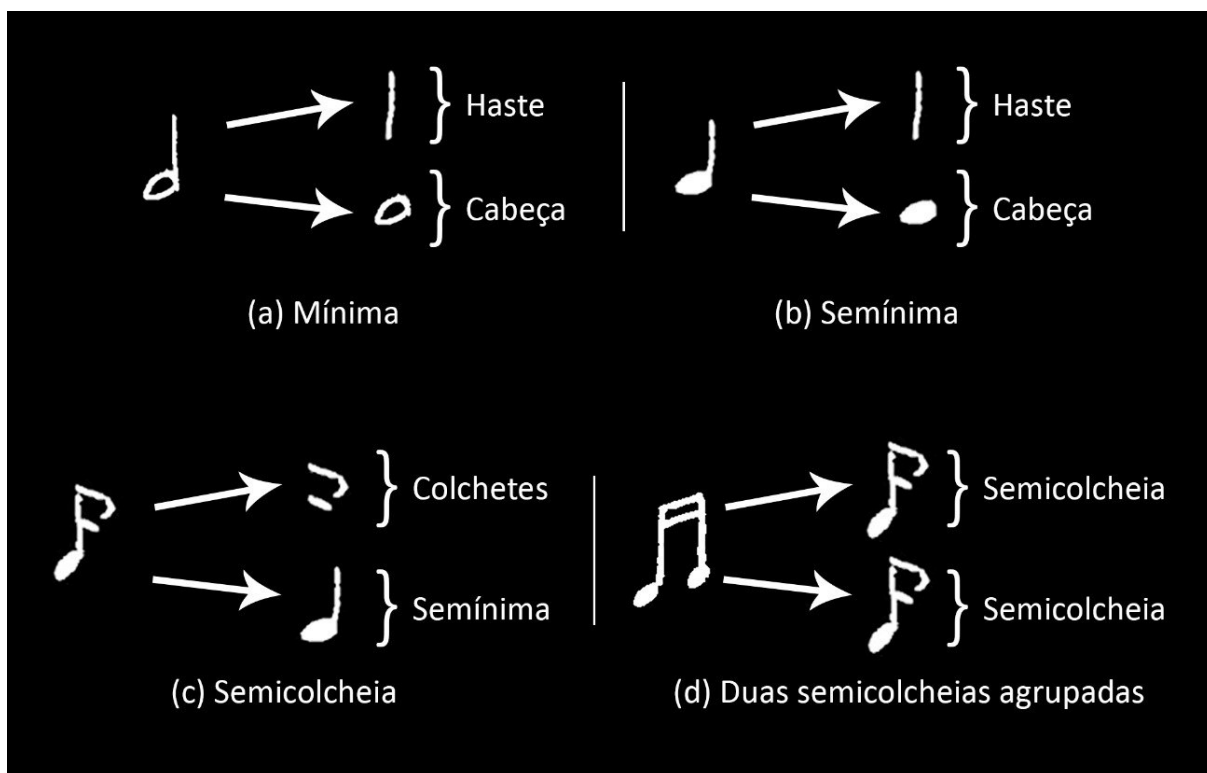
5.5 Reconstrução da Notação Musical

Outra questão a ser estudada refere-se a análise específica sobre as posições das notas na partitura. Quando trabalhadas sobre partituras com símbolos digitais, facilmente pode ser detectada a cabeça da nota e por sua vez identificada a sua posição, permitindo

a reconstrução da notação musical do símbolo selecionado. Entretanto, quanto tratamos com notas manuscritas as cabeças das notas não possuem estruturas são regulares, pois, um mesmo autor em uma mesma partitura representa a mesma nota de formas diferentes, além disso, existem outros obstáculos para a identificação, vejamos alguns desses, que foram enumerados por [Kato e Inokuchi \(1992\)](#).

- As notas possuem conexões com outras notas, além de muitas vezes apresentarem sobreposições e mesmo acentuada proximidade sobre outras notas, dificultando o processo de segmentação.
- A forma e o tamanho das notas não se apresentam de forma constante.
- As linhas de pauta muitas vezes interferem na estrutura das notas por estarem muitas vezes na mesma posição, ocasionando fragmentação dos símbolos quando as pautas são removidas.
- Alguns símbolos são ambíguos, bem como a relação semântica entre os símbolos, que também apresentam imprecisões.

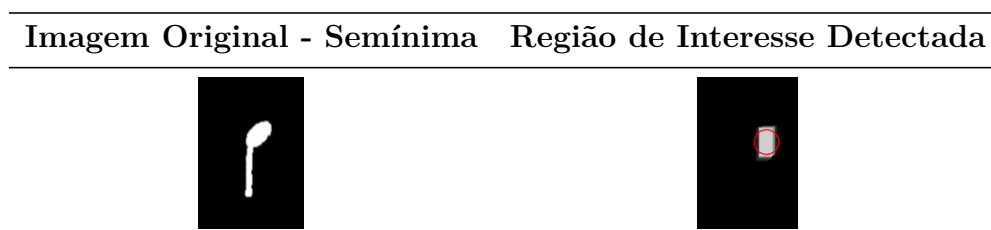
Figura 24 – Principais componentes da estrutura primitiva de uma nota musical, reconhecidas como: hastes, cabeças e colchetes.



Portanto, como tentativa, para solucionar alguns dos problemas acima, foi proposta a utilização da transformadas de Hough, para a detecção de segmentos de reta nas estruturas primitivas das notas musicais.

Na Figura 24, é possível reconhecer os elementos primitivos responsáveis pela construção de quase todas as notas musicais, esses são: as hastes, as cabeças e os colchetes. Portanto, se for efetuado a desagregação das formas primitivas, separando-as dos outros elementos, então torna-se possível a detecção, por exemplo, das cabeças das notas, permitindo assim reconhecer a sua posição em relação a sua linha de pauta. Logo, a remoção dos segmentos de reta, correspondentes as hastes e a aplicação de operações morfológicas de dilatação sobre um elemento estruturante na forma de elipse, torna possível a identificação das posições das notas na pauta.

Tabela 8 – Apresenta a imagem original da semínima e a imagem resultante com a região de interesse (cabeça) demarcada.



Na Tabela 8, temos um exemplo de remoção da haste para uma semínima, embora não possa ser considerada uma abordagem exaustiva e completa, por apresentar erros significativos, ainda assim, a metodologia retornou resultados bastante precisos, como poderá ser visto no resultado final da integração de todas as técnicas apresentadas.

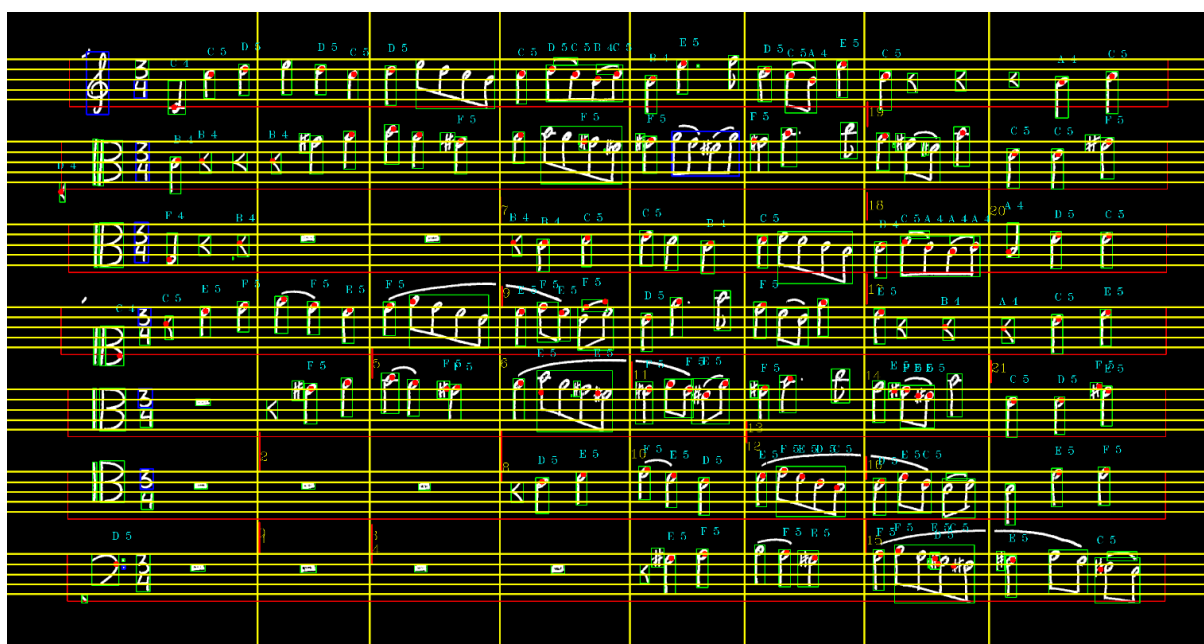
Na Figura 25, é possível visualizar, as linhas de pauta detectadas, bem como as linhas verticais que delimitam os compassos, que no caso acima estavam todos na mesma posição, também temos a região de interesse da pauta, identificada pelo quadrado vermelho que agrega cada pauta, que por sua vez, fornece ao sistema a identificação de todas as notas que pertencem a região da pauta.

Também, temos a identificação das claves, que por sua vez são necessárias para definir a altura das notas da sua pauta. E por fim, temos os símbolos musicais detectados, com a identificação da sua posição em relação a pauta sendo exibidos por cima das notas.

Nas Tabelas 9 e 10, temos os resultados mais detalhados para a detecção das regiões da partitura, como as linhas e barras de compasso, bem como, as partes gráficas primitivas que compõe as estruturas das notas, no caso, as cabeças das notas e barras de compasso. Nas quais, foram escolhidas cinco partituras da base de dados CVC-MUSCIMA com a mesma composição, mas de autores diferentes, para a realização desses testes específicos.

Foi possível notar, que embora a etapa de reconhecimento das linhas das pautas

Figura 25 – Versão final das informações que foram identificadas e classificadas na partitura manuscrita submetida a metodologia apresentada.



Fonte – Autor

Tabela 9 – Apresenta os dados para a detecção e remoção das pautas.

Partituras	Número de Pautas / Linhas	Pautas / Linhas		Pautas / Linhas	
		Detectadas	Percentual	Removidas Perfeitamente	Percentual
1	7 / 35	7 / 35	100%	7 / 35	100%
2	7 / 35	7 / 35	100%	6 / 34	97,14%
3	7 / 35	7 / 35	100%	7 / 35	100%
4	7 / 35	7 / 35	100%	6 / 33	94,28%
5	7 / 35	7 / 35	100%	7 / 35	100%

Tabela 10 – Apresenta os dados para a detecção das barras de compasso e cabeças das notas.

Partituras	Cabeça das Notas /		Barras de Compasso /	
	Percentual (VP)		Percentual (VP)	
1	203 / 222	91,44%	41 / 49	83,67%
2	198 / 222	89,18%	32 / 49	65,30%
3	159 / 222	71,62%	31 / 49	63,26%
4	212 / 222	95,49%	19 / 49	38,77%
5	184 / 222	82,88%	39 / 49	79,59%

tenha auferido 100% de acerto, a etapa subsequente de remoção não conseguiu propagar os efeitos da acurácia sobre todos os dados, pois no momento da extração das pautas ocorrem perdas de dados, ocasionando muitas vezes a segmentação de símbolos que estão

sobre as linhas do pentagrama.

Nas etapas posteriores, de detecção das barras de compasso e reconhecimento das cabeças notas, os métodos aplicados apresentaram taxas médias acima de 65% de acerto para os verdadeiros positivos, refletindo a necessidade da busca por metodologias mais precisas no reconhecimento de traços e círculos manuscritos, pois esses, correspondem as estruturas primitivas das notas e barras de compasso.

5.6 Publicação do arquivo digital da partitura

Também foi possível realizar a conversão das estruturas reconhecidas para o formato MusicXML. Criado pela Recordare LLC em 2004, o formato baseado em XML para a representação da notação musical ocidental, possui uma licença para o uso livre, embora o formato MusicXML seja proprietário, possuindo as suas definições de tipo de documento (DTD) muito bem documentadas, além de um esquema de validação (XSD) do W3C.

Sendo projetado como um formato de intercâmbio para aplicações de notação, análise, recuperação e desempenho. Pretendendo abarcar especificamente a notação musical ocidental a partir do século 17 em diante ([GOOD et al., 2001](#)).

Portanto, o formato MusicXML, permite assim, o intercâmbio das informações entre diversas aplicações disponíveis no mercado, assim como, soluções de código-aberto, de forma robusta e ampla, proporcionando o compartilhamento das músicas que foram recuperadas.

O arquivo de retorno, Figura 26, nos apresenta uma tentativa de recuperação com um acerto correspondente a aproximadamente 70% da partitura original, valor obtido devido as variações na notação musical, bem como classificações incorretas para as categorias de símbolos extraídos. Mas o que não deixa de ser um comportamento natural diante das questões apresentadas, sendo necessário a partir desse momento, a individualização e resolução para cada subproblema encontrado.

Foi possível, perceber a dimensão e complexidade quando tratamos do reconhecimento de partituras manuscritas, muitas técnicas consideradas estado da arte foram empregadas com sucesso, resultando em ótimo retorno de acurácia para certas etapas, mas que apenas nos permitem visualizar como o reconhecimento de um documento manuscrito, é complexo e demanda a utilização dos mais diversos métodos de análise e recuperação de informações.

Figura 26 – Partitura recuperada, em sua versão final, convertida do formato MusicXML gerado pelas metodologias abordadas nesse estudo.



6 Conclusão

Podemos afirmar, que a análise, reconhecimento e reconstrução de uma partitura manuscrita envolve inúmeras metodologias, empregadas nos mais diversos campos da computação, que prosseguem desde a criação de estruturas de dados, reconhecimento de imagens, até análises e classificações complexas sobre as informações recuperadas.

O algoritmo de extração de características ORB como solução para reconhecimento de notas musicais manuscritas ou para uma base de dados onde os objetos em questão apresentem formas bastante distintas apresentou-se bastante acurado, conseguindo extrair características relevantes para a criação do BoK e posterior classificação no SVM, mas falhando ao encontrar símbolos estruturalmente parecidos, o que não inviabiliza o resultado final, sobre a afirmação de que foi apresentado um método relevante que garante a detecção de objetos manuscritos, afirmação que pode ser confirmada na tabela de comparação de resultados com outras metodologias já empregadas.

Apesar dos reconhecidos avanços na detecção de símbolos musicais manuscritos, ainda existem muitos problemas em aberto que necessitam ser solucionados. Sendo, um dos mais básicos, a não existência de um repositório amplo de partituras manuscritas, que apresentem de fato os efeitos do: tempo, manuseio e armazenamento sobre o papel em foram impressos, permitindo a aplicação sobre dados reais, além de suprir a falta de amostras para compor as bases de treinamento. Também, existe a imprescindibilidade da estruturação de um framework robusto contendo as principais regras sintáticas e semânticas para o reconhecimento de notas e partituras musicais manuscritas.

Outro fator interessante a ser notado, refere-se à ambiguidade encontrada em diversos símbolos musicais, que com a adoção de algoritmos para extração de características, que sejam invariantes a escala e rotação, começam a demonstrar erros de classificação. Portanto, é indispensável novas pesquisas relacionadas ao comportamento de certos símbolos com estruturas semelhantes sobre abordagens invariantes a rotação, verificando a necessidade legítima, seja de substituição dos algoritmos conhecidos, ou mesmo no emprego de mecanismos em conjunto para permitir a mitigação das taxas de erros.

6.1 Trabalhos Futuros

Portanto, em estudos futuros propomos a utilização de técnicas conjuntas de extração e segmentação e pré-processamento de imagens, abordando métodos que possibilitem a extração características estruturais dos símbolos, como o Template Matching, bem como novas abordagens, utilizando os algoritmos de extração como o FREAK (Fast

Retina Keypoint) ou LATCH (Learned Arrangements of Three Patch Codes) para o reconhecimento tanto de notas musicais manuscritas bem como outros objetos manuscritos, como: os símbolos ortográficos. Além do emprego do método de ajuste de cisalhamento para resolver problemas de curvatura em pautas, assim como a utilização de métodos interativos no processo de reconhecimento de partituras manuscritas.

Referências

- BAINBRIDGE, D. Extensible optical music recognition. University of Canterbury. Computer Science, 1997. Citado na página 14.
- BAINBRIDGE, D.; BELL, T. The challenge of optical music recognition. *Computers and the Humanities*, Springer, v. 35, n. 2, p. 95–121, 2001. Citado na página 15.
- BARCELLOS, J. *Partituras para imprimir ou baixar*. Brasil: Ensino de Música, 2017. <<https://juarezbarcellos.files.wordpress.com/2013/02/hino-nacional-brasileiro-partitura-cifra-e-letra.pdf>>. Citado na página 24.
- BARZILAY, O.; BRAILOVSKY, V. L. On domain knowledge and feature selection using a support vector machine. *Pattern Recognition Letters*, Elsevier, v. 20, n. 5, p. 475–484, 1999. Citado na página 41.
- BELLINI, P.; BRUNO, I.; NESI, P. Assessing optical music recognition tools. *Computer Music Journal*, MIT Press, v. 31, n. 1, p. 68–93, 2007. Citado na página 15.
- BEN-DAYAN, B.; GILOH, I. Optical music recognition. Ben-Gurion University. Department of Electrical Engineering, 2013. Citado na página 14.
- BEN-NAIM, E.; FRAUENFELDER, H.; TOROCZKAI, Z. *Complex networks*. [S.l.]: Springer Science & Business Media, 2004. v. 650. Citado na página 42.
- BENNETT, R. *Uma breve história da música*. [S.l.]: Jorge Zahar Editora, Rio de Janeiro, 1986. Citado na página 23.
- BISHOP, C. M. *Pattern recognition and machine learning*. [S.l.]: springer, 2006. Citado na página 34.
- BLOSTEIN, D.; BAIRD, H. S. et al. A critical survey of music image analysis. *Structured Document Image Analysis*, Springer Verlag, p. 405–434, 1992. Citado na página 15.
- BRADSKI, G.; KAEHLER, A. *Learning OpenCV: Computer vision with the OpenCV library*. [S.l.]: "O'Reilly Media, Inc.", 2008. Citado na página 46.
- CAI, Y.-D.; LIU, X.-J.; XU, X.-b.; CHOU, K.-C. Prediction of protein structural classes by support vector machines. *Computers & chemistry*, Elsevier, v. 26, n. 3, p. 293–296, 2002. Citado na página 41.
- CALONDER, M.; LEPETIT, V.; STRECHA, C.; FUA, P. Brief: Binary robust independent elementary features. *Computer Vision–ECCV 2010*, Springer, p. 778–792, 2010. Citado 2 vezes nas páginas 35 e 37.
- CALVO-ZARAGOZA, J.; PERTUSA, A.; ONCINA, J. Staff-line detection and removal using a convolutional neural network. *Machine Vision and Applications*, Springer, p. 1–10, 2017. Citado na página 20.
- CHANDA, S.; DAS, D.; PAL, U.; KIMURA, F. Offline hand-written musical symbol recognition. In: IEEE. *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*. [S.l.], 2014. p. 405–410. Citado na página 18.

- CHAVES, A. d. C. F. Extração de regras fuzzy para máquinas de vetores suporte (svm) para classificação em múltiplas classes. *Rio de Janeiro*, 2006. Citado na página 43.
- CHEDIAK, A. *Harmonia e Improvisação-Vol. 1*. [S.l.]: Irmãos Vitale, 1986. Citado na página 23.
- CHOPIN, F. *Sonate pour piano et violoncelle op. 65*. France: NATIONAL LIBRARY OF FRANCE, 2017. <<http://blog.europeana.eu/2013/02/discover-chopins-handwritten-manuscripts>>. Citado na página 25.
- COMER, M. L.; DELP, E. J. Morphological operations for color image processing. *J. Electronic Imaging*, v. 8, n. 3, p. 279–289, 1999. Citado na página 33.
- CORTES, C.; VAPNIK, V. Support-vector networks. *Machine learning*, Springer, v. 20, n. 3, p. 273–297, 1995. Citado na página 41.
- CSURKA, G.; DANCE, C.; FAN, L.; WILLAMOWSKI, J.; BRAY, C. Visual categorization with bags of keypoints. In: PRAGUE. *Workshop on statistical learning in computer vision, ECCV*. [S.l.], 2004. v. 1, n. 1-22, p. 1–2. Citado na página 38.
- DALITZ, C. *Document Image Analysis with the Gamera Framework*. [S.l.]: Shaker Verlag GmbH, Germany, 2009. Citado na página 19.
- DALITZ, C.; DROETTBOOM, M.; PRANZAS, B.; FUJINAGA, I. A comparative study of staff removal algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 30, n. 5, p. 753–766, 2008. Citado 3 vezes nas páginas 19, 28 e 31.
- DALITZ, C.; KARSTEN, T.; POSE, F. Staff line removal toolkit for gamera. 2005. Citado na página 52.
- FLACH, P. A. The geometry of roc space: understanding machine learning metrics through roc isometrics. In: *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*. [S.l.: s.n.], 2003. p. 194–201. Citado na página 44.
- FORNÉS, A.; DUTTA, A.; GORDO, A.; LLADÓS, J. Cvc-muscima: a ground truth of handwritten music score images for writer identification and staff removal. *International Journal on Document Analysis and Recognition*, Springer, p. 1–9, 2012. Citado 2 vezes nas páginas 46 e 51.
- FORNÉS, A.; LLADÓS, J.; SÁNCHEZ, G. Primitive segmentation in old handwritten music scores. In: SPRINGER. *International Workshop on Graphics Recognition*. [S.l.], 2005. p. 279–290. Citado na página 33.
- FORNÉS, A.; SÁNCHEZ, G. Analysis and recognition of music scores. In: *Handbook of Document Image Processing and Recognition*. [S.l.]: Springer, 2014. p. 749–774. Citado na página 27.
- FREEMAN, H. On the encoding of arbitrary geometric configurations. *IRE Transactions on Electronic Computers*, IEEE, n. 2, p. 260–268, 1961. Citado na página 15.
- FUJINAGA, I. *Optical music recognition using projections*. Tese (Doutorado) — McGill University Montreal, Canada, 1988. Citado na página 20.

- FUJINAGA, I. Staff detection and removal. In: *Visual Perception of Music Notation: On-Line and Off Line Recognition*. [S.l.]: IGI Global, 2004. p. 1–39. Citado na página 31.
- GOOD, M. et al. Musicxml: An internet-friendly format for sheet music. In: *XML Conference and Expo*. [S.l.: s.n.], 2001. p. 03–04. Citado na página 66.
- GORDO, A.; FORNÉS, A.; VALVENY, E. Writer identification in handwritten musical scores with bags of notes. *Pattern Recognition*, Elsevier, v. 46, n. 5, p. 1337–1345, 2013. Citado 2 vezes nas páginas 19 e 31.
- GRANA, C.; BORGHESANI, D.; CUCCHIARA, R. Connected component labeling techniques on modern architectures. *Image Analysis and Processing–ICIAP 2009*, Springer, p. 816–824, 2009. Citado na página 34.
- GRANA, C.; BORGHESANI, D.; MANFREDI, M.; CUCCHIARA, R. A fast approach for integrating orb descriptors in the bag of words model. In: *Proc. SPIE*. [S.l.: s.n.], 2013. v. 8667, p. 866709–866709. Citado na página 40.
- GROUT, D.; PALISCA, C. *História da música ocidental*. [S.l.]: Gradiva, Rio de Janeiro, 1988. 21–31 p. Citado na página 23.
- HALL, M.; FRANK, E.; HOLMES, G.; PFAHRINGER, B.; REUTEMANN, P.; WITTEN, I. H. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, ACM, v. 11, n. 1, p. 10–18, 2009. Citado na página 46.
- HANUSIAK, R. K.; JUSTINO, E. J. R.; OLIVEIRA, L. S.; SABOURIN, R. Identificação da autoria de manuscritos com base em atributos genéticos e genéricos da escrita. *Revista de Informática Teórica e Aplicada*, v. 17, n. 2, p. 193–209, 2010. Citado na página 43.
- HARALICK, R. M.; STERNBERG, S. R.; ZHUANG, X. Image analysis using mathematical morphology. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, n. 4, p. 532–550, 1987. Citado na página 32.
- HOMENDA, W. Optical music recognition: the case study of pattern recognition. *Computer Recognition Systems*, Springer, p. 835–842, 2005. Citado na página 15.
- HSU, C.-W.; CHANG, C.-C.; LIN, C.-J. et al. A practical guide to support vector classification. Taipei, Taiwan, 2003. Citado na página 57.
- HSU, C.-W.; LIN, C.-J. A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, IEEE, v. 13, n. 2, p. 415–425, 2002. Citado na página 42.
- HU, M.-K. Visual pattern recognition by moment invariants. *IRE transactions on information theory*, IEEE, v. 8, n. 2, p. 179–187, 1962. Citado na página 61.
- ILLINGWORTH, J.; KITTLER, J. A survey of the hough transform. *Computer vision, graphics, and image processing*, Elsevier, v. 44, n. 1, p. 87–116, 1988. Citado na página 51.
- JOHANSEN, L. S. *Optical music recognition*. Dissertação (Mestrado), 2009. Citado na página 14.
- JONES, G.; ONG, B.; BRUNO, I.; KIA, N. Optical music imaging: music document digitisation, recognition, evaluation, and restoration. In: *Interactive multimedia music technologies*. [S.l.]: IGI Global, 2008. p. 50–79. Citado 2 vezes nas páginas 15 e 27.

- JR, J. H.; PECINA, P. Detecting noteheads in handwritten scores with convnets and bounding box regression. *arXiv preprint arXiv:1708.01806*, 2017. Citado na página 20.
- KASSLER, M. Optical character-recognition of printed music: A review of two dissertations. JSTOR, 1972. Citado na página 15.
- KATO, H.; INOKUCHI, S. A recognition system for printed piano music using musical knowledge and constraints. *Structured Document Image Analysis*, Springer Berlin Heidelberg, p. 435–455, 1992. Citado na página 63.
- KAVALLIERATOU, E.; STATHIS, S. Adaptive binarization of historical document images. In: IEEE. *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*. [S.l.], 2006. v. 3, p. 742–745. Citado na página 28.
- KILMER, A. D. Old babylonian musical instructions relating to hymnody. *Journal of Cuneiform Studies*, JSTOR, v. 38, n. 1, p. 94–98, 1986. Citado na página 22.
- KLINDWORTH, K. *Oeuvres complètes de Frédéric Chopin. Preludes, Op.28 (Chopin, Frédéric)*. USA: IMSLP, 2017. <[http://imslp.org/wiki/Preludes,_Op.28_\(Chopin,_Frédéric\)](http://imslp.org/wiki/Preludes,_Op.28_(Chopin,_Frédéric))>. Citado na página 25.
- KOHAVI, R. et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: STANFORD, CA. *Ijcai*. [S.l.], 1995. v. 14, n. 2, p. 1137–1145. Citado na página 56.
- KOLOMOETS, D. *Como o monge inventou a música*. Ucrânia: Collegium Musicum, 2017. <<http://collegiummusicum.com.ua/yak-chernets-vynajshov-muzyku>>. Citado na página 23.
- LOWE, D. G. Object recognition from local scale-invariant features. In: IEEE. *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. [S.l.], 1999. v. 2, p. 1150–1157. Citado na página 36.
- MACQUEEN, J. et al. Some methods for classification and analysis of multivariate observations. In: OAKLAND, CA, USA. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. [S.l.], 1967. v. 1, n. 14, p. 281–297. Citado na página 39.
- MATHERON, G. *Eléments pour une théorie des milieux poreux*. Masson, 1967. Citado na página 32.
- MAYORAZ, E.; ALPAYDIN, E. Support vector machines for multi-class classification. *Engineering Applications of Bio-Inspired Artificial Neural Networks*, Springer, p. 833–842, 1999. Citado na página 41.
- MELCHIORRE, C.; MATTEUCCI, M.; AZZONI, A.; ZANCHI, A. Artificial neural networks and cluster analysis in landslide susceptibility zonation. *Geomorphology*, Elsevier, v. 94, n. 3, p. 379–400, 2008. Citado na página 40.
- MIYAO, H.; MARUYAMA, M. An online handwritten music score recognition system. In: IEEE. *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. [S.l.], 2004. v. 1, p. 461–464. Citado na página 15.

- NG, K. Embracing the composer: Optical recognition of handwritten manuscripts. In: *Proc. of International Computer Music Conference*. [S.l.: s.n.], 1999. p. 500–503. Citado na página 33.
- NOSAL, E.-M. Flood-fill algorithms used for passive acoustic detection and tracking. In: IEEE. *New Trends for Environmental Monitoring Using Passive Systems, 2008*. [S.l.], 2008. p. 1–5. Citado na página 34.
- OH, J.; SON, S. J.; LEE, S.; KWON, J.-W.; KWAK, N. Online recognition of handwritten music symbols. *International Journal on Document Analysis and Recognition (IJDAR)*, Springer, v. 20, n. 2, p. 79–89, 2017. Citado na página 18.
- OTSU, N. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, IEEE, v. 9, n. 1, p. 62–66, 1979. Citado 2 vezes nas páginas 29 e 50.
- OTTEN, J. The catholic encyclopedia: An international work of reference on the constitution, doctrine, discipline and history of the catholic church. In: ROBERT APPLETON COMPANY. *HERBERMANN, C. G. et al. (Ed.)*. New York, 1910. p. 65–66. Citado na página 22.
- PEREIRA, R. M. P.; MATOS, C. E.; JUNIOR, G. B.; ALMEIDA, J. D. de; PAIVA, A. C. de. A deep approach for handwritten musical symbols recognition. In: ACM. *Proceedings of the 22nd Brazilian Symposium on Multimedia and the Web*. [S.l.], 2016. p. 191–194. Citado na página 61.
- PREARU, D. S. Computer pattern recognition of standard engraved music notation. *Ph. D thesis, Massachusetts Institute of Technology*, 1970. Citado na página 15.
- PRUSLIN, D. H. Automated recognition of sheet music. Ph. D thesis, Massachusetts Institute of Technology, 1966. Citado na página 15.
- PUGIN, L. Optical music recognition of early typographic prints using hidden markov models. In: *ISMIR*. [S.l.: s.n.], 2006. p. 53–56. Citado na página 19.
- REBELO, A.; CAPELA, G.; CARDOSO, J. S. Optical recognition of music symbols. *International journal on document analysis and recognition*, Springer, v. 13, n. 1, p. 19–31, 2010. Citado na página 15.
- REBELO, A.; FUJINAGA, I.; PASZKIEWICZ, F.; MARCAL, A. R.; GUEDES, C.; CARDOSO, J. S. Optical music recognition: state-of-the-art and open issues. *International Journal of Multimedia Information Retrieval*, Springer, v. 1, n. 3, p. 173–190, 2012. Citado 4 vezes nas páginas 15, 27, 28 e 50.
- REBELO, A. M. Robust optical recognition of handwritten musical scores based on domain knowledge. *Document Analysis and Recognition*, v. 13, p. 19–31, 2012. Citado 5 vezes nas páginas 18, 27, 31, 45 e 49.
- RILEY, J.; FUJINAGA, I. Recommended best practices for digital image capture of musical scores. *OCLC Systems & Services: International digital library perspectives*, MCB UP Ltd, v. 19, n. 2, p. 62–69, 2003. Citado na página 28.
- ROSIN, P. L. Measuring corner properties. *Computer Vision and Image Understanding*, Elsevier, v. 73, n. 2, p. 291–307, 1999. Citado na página 37.

- ROSTEN, E.; PORTER, R.; DRUMMOND, T. Faster and better: A machine learning approach to corner detection. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 32, n. 1, p. 105–119, 2010. Citado na página 36.
- ROY, P. P.; BHUNIA, A. K.; PAL, U. Hmm-based writer identification in music score documents without staff-line removal. *Expert Systems with Applications*, Elsevier, v. 89, p. 222–240, 2017. Citado 2 vezes nas páginas 19 e 31.
- RUBLEE, E.; RABAUD, V.; KONOLIGE, K.; BRADSKI, G. Orb: An efficient alternative to sift or surf. In: IEEE. *Computer Vision (ICCV), 2011 IEEE international conference on*. [S.l.], 2011. p. 2564–2571. Citado na página 35.
- SAUVOLA, J.; SEPPANEN, T.; HAAPAKOSKI, S.; PIETIKAINEN, M. Adaptive document binarization. In: IEEE. *Document Analysis and Recognition, 1997., Proceedings of the Fourth International Conference on*. [S.l.], 1997. v. 1, p. 147–152. Citado na página 29.
- SERRA, J. *Image analysis and mathematical morphology, v. 1*. [S.l.]: Academic press, 1982. Citado na página 32.
- SUYKENS, J. A.; VANDEWALLE, J. Multiclass least squares support vector machines. In: IEEE. *Neural Networks, 1999. IJCNN'99. International Joint Conference on*. [S.l.], 1999. v. 2, p. 900–903. Citado na página 42.
- SUZUKI, S. et al. Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, Elsevier, v. 30, n. 1, p. 32–46, 1985. Citado na página 34.
- TORBERT, S. *Applied computer science*. [S.l.]: Springer, 2016. 146–149 p. Citado na página 34.
- TRIER, O. D.; TAXT, T. Evaluation of binarization methods for document images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 17, n. 3, p. 312–315, 1995. Citado na página 28.
- VAPNIK, V. Principles of risk minimization for learning theory. In: *Advances in neural information processing systems*. [S.l.: s.n.], 1992. p. 831–838. Citado na página 42.
- VAPNIK, V. N.; CHERVONENKIS, A. Y. On the uniform convergence of relative frequencies of events to their probabilities. In: *Measures of complexity*. [S.l.]: Springer, 2015. p. 11–30. Citado na página 43.
- VAPNIK, V. N.; VAPNIK, V. *Statistical learning theory*. [S.l.]: Wiley New York, 1998. v. 1. Citado na página 41.
- YADID-PECHT, O.; GERNER, M.; DVIR, L.; BRUTMAN, E.; SHIMONY, U. Recognition of handwritten musical notes by a modified neocognitron. *Machine vision and applications*, Springer, v. 9, n. 2, p. 65–72, 1996. Citado na página 15.