

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

Investigação e Aperfeiçoamento de Algoritmos de Reconhecimento de Símbolos Musicais

Andreas Dieter Mendes Seufert

Dissertação realizada no âmbito do Mestrado Integrado em Engenharia Electrotécnica e
de Computadores Major Telecomunicações

Orientador: Prof. Dr. Jaime dos Santos Cardoso

Responsável no INESC Porto: Mestre Ana Rebelo

Janeiro 2010

A Dissertação intitulada

**“INVESTIGAÇÃO E APERFEIÇOAMENTO DE ALGORITMOS DE RECONHECIMENTO DE
SÍMBOLOS MUSICAIS”**

foi aprovada em provas realizadas em 25/Fevereiro/2010

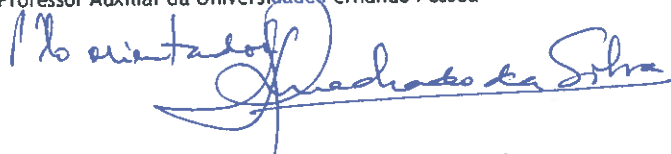
o júri



Presidente Professor Doutor Artur Pimenta Alves
Professor Catedrático do Departamento de Engenharia Electrotécnica e de Computadores da
Universidade do Porto



Professor Doutor José Manuel de Castro Torres
Professor Auxiliar da Universidade Fernando Pessoa



Professor Doutor Jaime dos Santos Cardoso
Professor Auxiliar do Departamento de Engenharia Electrotécnica e de Computadores da
Faculdade de Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projecto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extractos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são correctamente citados.



AUTOR: ANDREAS DIETER MENDES SEUFERT

Faculdade de Engenharia da Universidade do Porto

Resumo

Partituras musicais antigas apenas se encontram disponíveis em versões manuscritas ou em fotocópias. Ao longo dos anos tem-se perdido muita cultura musical devido à má preservação daquelas. O projecto de reconhecimento óptico de partituras musicais, em desenvolvimento no INESC Porto, *Optical music recognition* (OMR), visa, como objectivo primordial, a preservação da cultura musical, principalmente a nível nacional.

O processo de reconhecimento de partituras musicais manuscritas é muito dispendioso em tempo e susceptível a erros, quando realizado manualmente. O OMR clássico está mais focado em pautas impressas e regulares. Contudo, neste trabalho o objecto de estudo são as partituras manuscritas e irregulares, para as quais as soluções actuais se encontram longe do ideal. Desenvolver uma técnica de OMR que possa, de forma automática ou semiautomática, representar uma pauta manuscrita em formato digital seria extremamente benéfico pois permitiria um acesso generalizado a partituras que nunca foram publicadas e, portanto, dificilmente acessíveis de momento.

A detecção de símbolos musicais em partituras manuscritas é um ponto crucial para um bom desempenho de um OMR. Neste projecto são testados os desempenhos de classificadores para a detecção de símbolos musicais em vários cenários. Podem tratar-se de símbolos sintéticos com deformações, de mistura de símbolos manuscritos com sintéticos ou de símbolos de autores separados. Também foi ensaiado um classificador menos convencional neste tipo de projecto. O reconhecimento de símbolos musicais em partituras manuscritas não tem sido alvo de investigação, merecendo desta forma um estudo aprofundado.

No decorrer deste trabalho foi possível analisar o desempenho de vários classificadores a vários tipos de classes, obtendo-se resultados positivos e promissores nalguns cenários e resultados inesperados noutros. Foi também introduzido na área do OMR o classificador máquina de vectores relevantes (RVM), uma solução alternativa às técnicas mais convencionais em OMR. Efectuou-se um estudo alargado de diversos algoritmos em diversos cenários, concluindo-se que as SVMs demonstram de um modo geral os resultados mais robustos que os restantes classificadores.

Abstract

Old musical sheets are only available in a handwritten style or in photocopies. A lot of musical culture was lost during the years because of the bad preservation of those. The project of optical recognition of handwritten musical sheets, in development in INESC Oporto, optical music recognition (OMR) has, as its principal goal, the preservation of musical heritage, specially the at a national basis.

The process of handwritten musical sheets recognition is very time consuming and susceptible to errors when done manually. The classic OMR is more focused in printed and regular music scores. However, in this work the objective of study are the handwritten and irregular music sheets, for which the actual solutions are far from ideal. A development of an OMR technique that, automatically or semi-automatically, represents a handwritten musical score in digital format would be extremely benefic, because it would allow a general access to musical sheets that have never been published and, due to that, have a hard access nowadays.

The detection of musical symbols in handwritten music scores is a crucial point for a good performance of an OMR. In this project the performance of classifiers for the detection of musical symbols in various scenarios are tested. There can be deformed synthetic symbols, a mixture of handwritten and synthetic symbols or symbols of different authors. A classifier that is less conventional in this kind of project was also tested. The recognition of musical symbols in handwritten sheets hasn't been a target of investigation, deserving a deeper study.

During this work the performance of various classifiers was analyzed and positive and promising results were achieved as well as unexpected ones. In the field of OMR the classifier "Relevance Vector Machine"(RVM) was introduced, as an alternative solution to the standard techniques. A vast study of various algorithms in different scenarios was done, with the SVMs showing, in a general view, the most robust results in comparison to the other classifiers.

Agradecimentos

O trabalho exaustivo e pesado que foi efectuado ao longo dos últimos meses não teria sido realizável sem o apoio de numerosas pessoas.

Quero agradecer em primeiro lugar ao meu orientador e professor Jaime Cardoso pela oportunidade de investigação no INESC Porto, pelo apoio durante a execução deste trabalho e pelos conselhos sensatos que levaram ao melhoramento deste. Também quero agradecer à responsável no INESC Porto, Ana Rebelo, pelas "horas extraordinárias" que teve de fazer para apoiar o meu trabalho e responder com muita paciência às minhas demais dúvidas que se foram acumulando no decorrer do projecto.

Quero também agradecer ao grupo de doutoramento e de mestrado orientado pelo Prof. Jaime Cardoso, pelo ambiente que forneceram e pelas "dicas" que iam dando sempre que observavam o meu trabalho, quer nas horas de trabalho quer em situações mais sociais.

Um agradecimento especial à minha mãe que sempre me apoiou e acreditou no meu trabalho e sempre teve tempo para uma ajuda, não só durante o tempo da elaboração desta tese, mas também em todo o meu percurso académico.

Ainda desejo acrescentar um agradecimento muito especial a uma pessoa também muito especial que me acompanhou com muito carinho e muita paciência nesta jornada, apoiando-me nas alturas mais críticas e oferecendo-me sempre motivação para continuar o trabalho com um sorriso. Muito obrigado pela constante presença ao meu lado!

Andreas Seufert
Fevereiro, 2010

“Even if, in one or other of them, I had a particular word or words in mind, I would not tell anyone, because the same word means different things to different people. Only the songs say the same thing, arouse the same feeling, in everyone - a feeling that can’t be expressed in words.”

“Mesmo que eu tivesse uma ou várias palavras na minha mente, eu não contaria a ninguém, pois a mesma palavra tem significados diferentes para pessoas diferentes. Apenas as músicas dizem o mesmo, elevam o mesmo sentimento em toda a gente - um sentimento que não pode ser expresso em palavras.”

Felix Mendelssohn, compositor alemão (1809-1847)

Ao meu pai, que nos deixou com muitas saudades, e à minha mãe

Conteúdo

1	Introdução	1
1.1	Introdução à Dissertação	1
1.2	Arquitectura de um OMR	3
1.3	Objectivos	4
1.4	Estrutura	4
1.5	Contribuições para o Projecto e Publicações Relacionadas	5
2	Estudo do Estado da Arte e Fundamentos Relevantes	7
2.1	Introdução	7
2.2	Análise do Estado da Arte	8
2.2.1	Detecção e Remoção de Linhas	8
2.2.2	Segmentação	9
2.2.3	Reconhecimento e Detecção	10
2.2.4	Esquema <i>Web</i> Existente	12
2.2.5	Resumo	12
2.3	Fundamentos Relevantes	12
2.3.1	Máquinas de Vector de Suporte (SVMs)	13
2.3.2	Redes Neurais	14
2.3.3	K-Vizinho Mais Próximo (k-Nearest Neighbour)	15
2.3.4	Máquinas de Vector de Relevância (RVMs)	15
3	Bases de Dados e Testes Realizados	19
3.1	Criação de Bases de Dados	19
3.2	Realização de Testes	21
3.2.1	Especificação nos Classificadores	22
3.2.2	Separação de Autores	22
3.2.3	Escolha de Características	22
4	Resultados e Análise	25
4.1	Extracção de Características	25
4.2	Resultados e Análise	26
4.2.1	Combinação de Partituras - Análise Apenas aos Símbolos	27
4.2.2	Deformações	28
4.2.3	Separação por Autores - Partituras Sintéticas	30
4.2.4	Separação por Autores - Partituras Manuscritas	32
4.2.5	Extracção de Características	33
4.2.6	Fusão de Extracção de Características e de Classificadores	34
4.2.7	Comparação de RVMs com SVMs	35

4.3	Análise Geral	37
5	Conclusões e Trabalho Futuro	39
5.1	Conclusões	39
5.2	Trabalho Futuro	40
A	Deformações	41
A.1	Deformações segundo <i>Kanungo</i>	41
A.2	Deformações <i>white speckles</i> (manchas brancas)	41
B	Cálculo do Intervalo de Confiança	43
C	Artigo Submetido a Conferência	45
	Referências	52

Lista de Figuras

1.1	Arquitetura genérica do sistema. Figura retirada de [1].	2
1.2	Estrutura típica de um <i>OMR</i> . Figura retirada de [2].	3
2.1	Diversidade de apresentação de símbolos musicais.	9
3.1	Diferentes tipos de classes.	19
3.2	Sustenido com vários tipos de deformações: (a) <i>white speckles</i> com $w=0.03$, (b) <i>white speckles</i> com $w=0.05$, (c) <i>white speckles</i> com $w=0.07$, (d) <i>white speckles</i> com $w=0.09$, (e) <i>white speckles</i> com $w=0.11$ e (f) deformação segundo <i>Kanungo</i> com $b=1$ e $a=0.5$	21
4.1	Exemplo de diferença de notas entre autores diferentes.	33

Lista de Tabelas

4.1	Resultados dos classificadores aos testes com base de dados de combinação de partituras.	27
4.2	Percentagem de detecção dos classificadores com base de dados de combinação de partituras.	27
4.3	Resultados dos classificadores aos testes com base de dados de símbolos sintéticos com deformações ligeiras.	28
4.4	Percentagem de detecção dos classificadores com base de dados de símbolos sintéticos com deformações ligeiras.	28
4.5	Resultados dos classificadores aos testes com base de dados de símbolos sintéticos com deformações acentuadas.	29
4.6	Percentagem de detecção dos classificadores com base de dados de símbolos sintéticos com deformações acentuadas.	30
4.7	Resultados dos classificadores aos testes com base de dados de símbolos sintéticos com deformações do tipo <i>white speckles</i> ligeiras e deformações segundo <i>Kanungo</i>	30
4.8	Percentagem de detecção dos classificadores com base de dados de símbolos sintéticos com deformações do tipo <i>white speckles</i> ligeiras e deformações segundo <i>Kanungo</i>	31
4.9	Resultados dos classificadores aos testes com a base de dados de partituras sintéticas separadas por autores.	32
4.10	Resultados dos classificadores aos testes com a base de dados de partituras manuscritas separadas por autores.	32
4.11	Resultados dos classificadores analisando apenas as características extraídas, com a base de dados de combinação de partituras.	33
4.12	Percentagem de detecção dos classificadores analisando apenas as características extraídas, com a base de dados de combinação de partituras.	34
4.13	Resultados dos classificadores fusionando os símbolos com as características extraídas com base de dados de combinação de partituras.	34
4.14	Percentagem de detecção dos classificadores fusionando os símbolos com as características extraídas com base de dados de combinação de partituras.	35
4.15	Resultados das RVMs e SVMs com base de dados de combinação de partituras.	35
4.16	Percentagem de detecção dos classificadores com base de dados de combinação de partituras.	36
4.17	Resultados das RVMs e SVMs com base de dados de partituras manuscritas.	36
4.18	Percentagem de detecção dos classificadores com base de dados de partituras manuscritas.	36
4.19	Resultados das RVMs e SVMs com base de dados de partituras sintéticas.	37
4.20	Percentagem de detecção dos classificadores com base de dados de partituras sintéticas.	37

Abreviaturas e Símbolos

HMM	Hidden Markov Model
INESC Porto	Instituto de Engenharia de Sistemas e Computadores do Porto
LAG	Line Adjacency Graph
OMR	Optical Music Recognition
OMRSYS	Optical Music Recognition System
RLC	Run-Length-Coding
RVM	Relevance Vector Machine
SVM	Support Vector Machine
XML	Extensible Markup Language

Capítulo 1

Introdução

Neste capítulo far-se-á uma introdução ao trabalho desenvolvido no âmbito desta dissertação. Deste modo, será efectuada não só uma introdução geral à tese, explicitando-se a motivação que conduziu à execução deste trabalho, mas também serão apresentados os objectivos propostos, o modo de definição da contextualização e a especificação da estrutura do presente relatório.

1.1 Introdução à Dissertação

Música, derivado do grego *μουσική (τέχνη)* – *musiké (téchne)*, a arte das musas, é uma forma de arte que consiste em combinar sons e silêncio seguindo ou não uma pré-organização ao longo do tempo. Em todas as culturas existem manifestações musicais próprias e é considerada como uma parte essencial da herança cultural de cada sociedade.

Embora nem sempre seja composta com esse objectivo, a música pode ser considerada como uma forma de arte, encarada por muitos como sua principal função. Mas também é utilizada para fins de educação, militares e de terapia. Como tal, a sua preservação em todas as suas formas tem que ser garantida.

Em Portugal existe uma enorme falha no que diz respeito à publicação musical em quase todas as áreas. Não há nenhum repositório de informação musical que tenha sido criado durante o último século. É por isso essencial combater esta falta de informação de modo a evitar uma perda irremediável da herança musical do país. A ferramenta mais utilizada para a preservação tem sido a digitalização de partituras, oferecendo possibilidades para duplicações, distribuições e processamento digital. No entanto, para que esta última funcionalidade possa ser fiável, é necessário um sistema de reconhecimento óptico musical (*Optical Music Recognition - OMR*), para se transformar as partituras e os manuscritos num formato passível de ser lido por máquinas. Infelizmente o estado da arte actual de reconhecimento de partituras manuscritas está longe de fornecer uma solução satisfatória.

O reconhecimento de notação musical através do computador, a sua interpretação e utilização entre várias aplicações levanta diversos desafios e questões em relação aos algoritmos apropriados, às técnicas e aos métodos com os quais se possa reconhecer a notação musical automaticamente.

A dissertação está inserida num projecto do INESC Porto (Instituto de Engenharia de Sistemas e Computadores), que tem como finalidade a criação de um sistema Web, que faça o reconhecimento óptico em formato XML (Extensible Markup Language) de partituras musicais manuscritas. Este projecto de reconhecimento automático, em curso desde 2007, no INESC Porto e na Escola Superior de Música e das Artes do Espectáculo (ESMAE), foi o início para a criação de um sistema OMR.

O objectivo deste projecto é ultrapassar o problema de reconhecimento de símbolos musicais em partituras manuscritas através da pesquisa e da aplicação das mais recentes técnicas em aprendizagem automática e inteligência artificial. Para além disto existe também a intenção de criar um sistema *web* que permita um acesso generalizado a um vasto número de folhas musicais não publicadas em formato digital. Esta base de dados não só centralizará a maior informação possível como também ajudará a preservar a herança musical de um modo inovador, com uma grande gama de possibilidades [3, 4]. A arquitectura proposta para o sistema está representada na Figura 1.1.

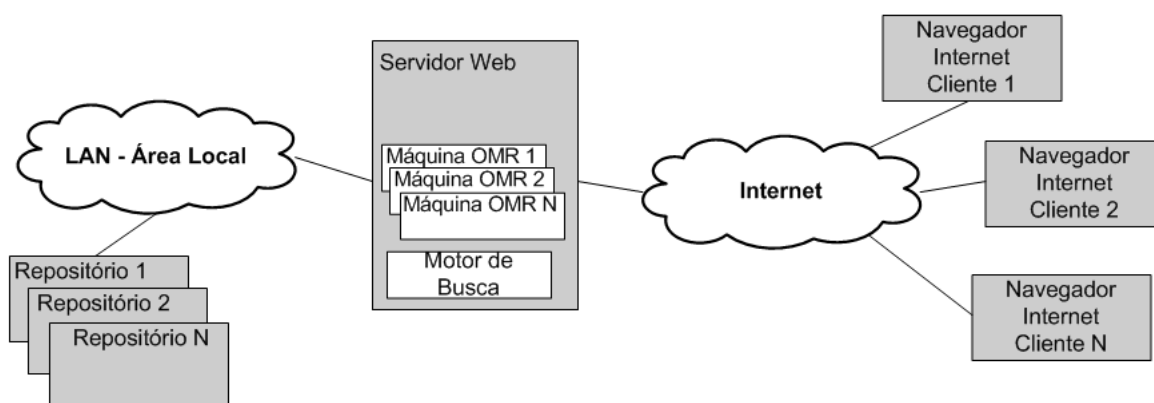


Figura 1.1: Arquitectura genérica do sistema. Figura retirada de [1].

O sistema é composto por três entidades diferentes: repositório, servidor *web* e *web browser*. Em suma, o módulo do repositório guarda a partitura original, o formato digital desta em *MusicXML* e todos os dados descritivos introduzidos pelo utilizador. A totalidade dos conteúdos restantes do sistema, tal como a informação do utilizador, são também gravados nesta entidade. O servidor *web* é o ponto de acesso do utilizador ao sistema, e a todos os seus módulos de processamento que correm no servidor, incorporando o motor de pesquisa e o motor de reconhecimento óptico para as partituras musicais. O servidor interage com o repositório e com *web browser*, que estabelece a interface entre o utilizador e o sistema. A interface do utilizador num *web browser* permite a gestão completa das partituras e dos dados associados, tal como a execução da administração do sistema.

O projecto aqui apresentado é inovador por diversas razões. Tem a ambição de desenvolver um formalismo para modelar de um modo consistente o conhecimento sobre linguagem e notação

musical. Explora a riqueza e o potencial das mais recentes técnicas de aprendizagem automática e inteligência artificial, não apenas para representar conhecimento mas também para tomar decisões. Pretende digitalizar e preservar um vasto corpo de partituras manuscritas numa forma sem precedentes.

Irá também incluir um motor OMR integrado com um sistema de arquivação e uma interface para pesquisa e edição. As partituras serão guardadas em *MusicXML*, um formato de interacção musical recente e em expansão desenhado para aplicações de notação, análise, recepção e desempenho. Tem ainda a intenção de possibilitar o acesso *online* aos repositórios das partituras manuscritas para fins de diversão, educação e musical.

1.2 Arquitectura de um OMR

Os principais objectivos de uma aplicação OMR são o reconhecimento, a representação e a gravação de partituras musicais num formato passível de ser lido por máquinas. Um programa OMR tem, então, que ser capaz de reconhecer o conteúdo musical e de fazer a análise semântica de cada símbolo musical de uma obra. No final, todas as informações musicais devem ser guardadas num formato de saída que seja facilmente lido por um computador.

De facto, a arquitectura de um sistema OMR é dependente dos métodos utilizados nas partes de segmentação e de reconhecimento.

A arquitectura típica de um OMR (Figura 1.2) consiste, numa fase inicial, de pré-processamento da imagem, o qual inclui a aplicação de várias técnicas, como binarização ou remoção de ruído, passando, de seguida, para a detecção dos símbolos musicais de uma partitura musical. Neste módulo, para o símbolo poder ser reconhecido de uma maneira eficaz, são, inicialmente, removidas as linhas da partitura, seguindo-se a segmentação do símbolo e, por fim, a classificação deste. O OMR, após o reconhecimento dos símbolos, efectua a reconstrução da notação musical e por fim a construção da representação final como uma descrição simbólica da partitura.



Figura 1.2: Estrutura típica de um OMR. Figura retirada de [2].

Os módulos de reconstrução da notação musical e representação final estão interligados intrinsecamente. Na reconstrução as primitivas dos símbolos são unidas de modo a formarem símbolos musicais. Neste passo aplicam-se, geralmente, regras gráficas e sintácticas. Isto é importante para a introdução de informação do contexto para validar e resolver ambiguidades do último módulo do reconhecimento de símbolos musicais. É também produzido um documento no qual os símbolos detectados são interpretados para serem associados a um sentido musical.

No módulo da representação final este documento é convertido num formato de descrição musical, tal como *MusicXML*, que permite a sua gravação.

Esta dissertação está centrada no passo de reconhecimento de símbolos musicais.

1.3 Objectivos

De modo a poder fazer-se um bom reconhecimento de símbolos musicais existentes em partituras musicais, serão treinados vários classificadores para conseguir testar a sua robustez e decidir qual o mais eficiente a reconhecer os símbolos. Para tal, extrair-se-ão símbolos musicais de partituras manuscritas e sintéticas com deformações para avaliar o desempenho de classificadores. Estes serão as redes neuronais, o k-vizinho mais próximo, as máquinas de vector de suporte (SVM - Support Vector Machines) e as máquinas de vector de relevância (RVM - Relevance Vector Machine).

Testar-se-ão os símbolos em diferentes cenários: misturas de símbolos de partituras manuscritas e de partituras sintéticas; símbolos com diferentes tipos de deformações. Será também feito um teste com os símbolos de autores diferentes, sempre para se tentar verificar o desempenho dos classificadores que estão a ser utilizados.

Serão confrontadas diferentes maneiras para se efectuar o reconhecimento de símbolos musicais, de modo a poderem ser confrontadas metodologias distintas.

No fim, com base nos resultados, será feita uma análise sucinta sobre a qualidade de cada um dos classificadores, para determinar qual o mais robusto para utilizar num trabalho futuro, isto é, no reconhecimento dos símbolos extraídos automaticamente das partituras.

Um outro objectivo é a inserção de uma metodologia nova para efectuar o reconhecimento de símbolos musicais. Até agora, o reconhecimento é sempre feito através da análise dos pixels do símbolo em si. Será agora efectuado e testado o reconhecimento através da extracção de diversas características para cada classe em análise.

Por fim, testar-se-á um classificador recente para possibilitar a comparação do seu desempenho em relação aos mais utilizados na literatura actual. O principal objectivo é o de alcançar resultados iguais, ou melhores, em menos tempo de computação comparativamente com os melhores algoritmos existentes.

1.4 Estrutura

O relatório está estruturado em 5 capítulos que descrevem as actividades realizadas no último semestre. O primeiro serve para introduzir o trabalho efectuado, tendo o segundo capítulo como conteúdo uma vasta análise do estado da arte e uma visão sobre os classificadores utilizados. No terceiro capítulo é explicado o trabalho que foi realizado no âmbito deste projecto e são descritos os testes efectuados, sendo apresentados os seus resultados e feita a sua análise no quarto capítulo. No quinto serão tiradas as conclusões referentes ao trabalho feito durante o último semestre e propostas tarefas para a continuação desta área.

1.5 Contribuições para o Projecto e Publicações Relacionadas

Esta dissertação apresenta as seguintes contribuições para a preservação e o acesso geral para a herança cultural e musical:

- Análise de desempenho de algoritmos de reconhecimento de símbolos musicais em partituras manuscritas;
- novos métodos e algoritmos para o reconhecimento e detecção automática de símbolos musicais.

Do trabalho relacionado com o projecto à qual esta dissertação pertence, espera o resultado da submissão de:

- "*Analysis of Classification Methods towards Musical Symbols Recognition*", Andreas Mendes Seufert, Ana Rebelo, André Marçal e Jaime S. Cardoso, em *Inductive Logic Programming - ILP 2010*

Capítulo 2

Estudo do Estado da Arte e Fundamentos Relevantes

Neste capítulo mostra-se o estado da arte e os fundamentos relevantes. Na primeira secção é efectuada uma pequena introdução. Na segunda apresentam-se as várias ideias e os métodos utilizados até hoje, de modo a tornar exequível o seu enquadramento neste projecto e justificar o ponto de partida deste trabalho dentro do projecto de reconhecimento automático de partituras musicais. Na terceira secção descrevem-se os classificadores que serão utilizados na fase de testes

2.1 Introdução

Várias técnicas de OMR foram desenvolvidas ao longo dos últimos dois anos, também aqui no INESC Porto. Guilherme Capela descreve na sua tese [5] um sistema automático de reconhecimento de pautas musicais, continuando um ano mais tarde no desenvolvimento de reconhecimento de símbolos musicais no *Framework gamera* [6]. Já Rebelo, Capela e Cardoso [2] apresentam no seu artigo um estudo comparativo de vários algoritmos de reconhecimento de símbolos musicais. Fizeram uma vasta revisão dos métodos mais usados neste contexto, avaliando também o seu rendimento. Rebelo [7] faz uma apresentação do estado da arte do reconhecimento óptico de partituras musicais baseado em fusões de regras musicais. Neste tema, Rossant e Bloch [8] também propõem um sistema robusto de um sistema OMR, baseado não só em fusões de regras musicais como também num algoritmo de detecção de erros.

Os classificadores utilizados ao longo deste trabalho foram as redes neuronais, o k-vizinho mais próximo, máquinas de vector de suporte e máquinas de vector de relevância. Este último, quase não utilizado em projectos semelhantes, foi utilizado, de acordo com estudos noutras áreas, com a expectativa de apresentar sensivelmente o mesmo desempenho que as SVMs, mas com menos tempo de treino e teste.

De modo a tentar reduzir o tempo de computação dos classificadores e tentar encontrar mais um método de classificar símbolos, também é efectuada uma extracção de características dos símbolos musicais. O estado da arte neste estudo será explicado ainda nesta secção.

2.2 Análise do Estado da Arte

2.2.1 Detecção e Remoção de Linhas¹.

No que diz respeito ao reconhecimento de símbolos musicais, o sistema típico de reconhecimento óptimo é o descrito na Figura 1.2, utilizado pela maior parte dos autores.

A importância da detecção e da remoção das linhas das partituras é a de permitir analisar mais facilmente os símbolos musicais, ficando estes isolados após a remoção das linhas. A maior parte das abordagens descritas na literatura começam com este procedimento. Mas se se considerar partituras manuscritas, a remoção complica-se bastante. Muitas das linhas não se encontram em linhas rectas, apresentam lacunas, não são paralelas entre si e, devido à idade dos manuscritos, manifestam um decaimento de qualidade em relação ao papel e à tinta. Muitos compositores têm, também, a sua maneira própria de escrever as partituras musicais, afastando-se de uma escrita normalizada.

O processo mais simples consiste em encontrar o máximo local na projecção horizontal dos pixels pretos da imagem [10, 11] ou a projecção vertical [12] pela transformada de Hough [7]. É possível efectuar várias projecções na horizontal através da rotação da imagem, escolhendo-se, por fim, a que tem o máximo local mais elevado. Assim, elimina-se o facto da suposição de todas as linhas serem horizontais.

A detecção das linhas baseada em *line adjacency graph* (LAG) [13] é uma estratégia alternativa. Este método procura potenciais secções das linhas: secções que satisfazem critérios relacionados com aspecto, conectividade e curvatura.

Outras técnicas de detecção de linhas baseiam-se em técnicas de processamento de imagem no algoritmo, incluindo *run-length-coding* (RLC) [10], análise de componentes conectados e de projecções. Depois de aplicar o RLC, para encontrar a grossura das linhas e o espaçamento entre estas, qualquer uma que tenha mais que o dobro e menos de metade da grossura das linhas será removida.

Também existem métodos que agrupam colunas verticais baseando-se no seu espaçamento, na sua grossura e posição vertical na imagem [14], sendo a classificação fundamentada em regras de linhas horizontais delgadas [15] e procura de linhas [16, 17].

Apesar da grande variedade de métodos existentes, todos têm bastantes limitações. As linhas com curvaturas ou descontinuidades nem sempre são detectadas. Existe um detector que trata descontinuidades [18]. Neste, a imagem é varrida pixel por pixel, encontrando regiões pretas que são classificadas como pequena região. Depois tenta unir essas regiões de modo a construir linhas.

¹Esta secção é baseada nos artigos [2, 9]

Um problema comum é que os métodos procuram construir linhas baseadas em informações locais, sem incorporarem informação global no processo. Nenhum deles tenta definir um processo que não se baseie apenas no facto de as linhas serem pretas e deixam de lado as suas informações globais. Mas existe uma proposta que soluciona este problema, na qual as linhas são soluções baseadas em processos de optimizações globais. O paradigma sugerido usa a imagem como um grafo, no qual as linhas resultam como caminhos conectados entre as duas margens da imagem [9]. Como as linhas são os únicos elementos pretos unidos de um lado ao outro da página de uma partitura musical, o caminho a procurar é o mais curto entre as duas margens. Segundo [2] este método é o que apresenta melhores resultados. Também tem a grande vantagem de não ser necessário como entrada o número correcto de linhas de pautas.

Se for efectuada uma boa detecção e remoção das linhas, é bastante mais simples detectar os símbolos musicais, pois apenas exige concentração nos símbolos em si e não em possíveis sobreposições.

Para a remoção das linhas Cardoso [2] modificou a versão de "*Line Track Height*" apresentado em [11] unindo este método com os resultados do processo dos caminhos estáveis (*stable path*). Consequentemente melhorou o algoritmo dando especial atenção a deformações - linhas de pautas podem ser descontínuas, curvadas ou inclinadas - que podem ocorrer em partituras musicais.

2.2.2 Segmentação

A segmentação de objectos musicais numa partitura musical tem sido alvo de pesquisa por uma grande comunidade. Mais uma vez deparamo-nos com problemas iguais aos dos de reconhecimento de linhas das partituras: degradação das folhas, falhas na digitalização e o facto de muitos dos símbolos a reconhecer saírem da notação clássica, reflectindo o estilo de escrita próprio de cada autor. Outro grande problema reside na diversidade existente nos símbolos impressos, como pode ser visto na figura 2.1.



Figura 2.1: Diversidade de apresentação de símbolos musicais.

O processo de segmentação consiste em localizar e isolar os símbolos de modo possibilitar a sua identificação correcta. Em [2] os símbolos são subdivididos em quatro categorias: símbolos

que estão apresentados com um segmento vertical maior que o *threshold* (notas, quer sejam abertas ou fechadas, como por exemplo a diferença entre mínima e semi-mínima); símbolos que unem notas (como é o caso das rectas que unem semicolcheias ou fusas); restantes símbolos unidos com as linhas de partituras (claves, pausas ou acidentes) e símbolos que se apresentam por cima ou por baixo das pautas (notas, relações e acentuações).

Na literatura, o processo de segmentação é normalmente baseado numa decomposição hierárquica de uma imagem musical. A partitura é analisada e decomposta em pautas - conjunto de cinco linhas - para posteriormente serem removidas as linhas e identificados os componentes ligados. Numa fase seguinte, são extraídos os símbolos.

2.2.3 Reconhecimento e Detecção

A detecção mais complexa é a das linhas horizontais que unem notas, devido à enorme variedade destes componentes existente na escrita de uma partitura. Neste sentido, em [2] estes símbolos são previamente removidos para facilitar a detecção das notas. Assim, o autor propõe um método baseado na forma geométrica e espacial dos objectos musicais.

A detecção de acidentes, de pausas e de acentuações é feita recorrendo a uma combinação da técnica de projecção de perfil X-Y ².

Em relação às claves, estas possuem atributos muito próprios para a detecção. Assim, em [7] foi tido em consideração o facto das claves serem os símbolos maiores e mais largos.

Para o processo de reconhecimento dos símbolos as diferentes abordagens basearam-se em três tipos [1]:

- Modelo de *Markov* escondido (HMM - *Hidden Markov Model*);
- redes neuronais;
- Máquinas de vectores de suporte (SVM - *Support Vector Machines*).

A autora, depois de realizados vários testes, concluiu que aquele que obteve melhores resultados para partituras musicais manuscritas foi o SVM. O modelo de *Markov* teve um desempenho melhor que redes neuronais visto o último ter um desvio padrão maior que o HMM e tem duas classes de símbolos que tiveram um resultado de 0% (claves de fá e notas abertas). Em relação às partituras impressas, os resultados que obtiveram com as SVMs são mais uma vez os resultados mais promissores.

O trabalho apresentado por [2] expõe um processo robusto para o reconhecimento automático para partituras musicais manuscritas. O algoritmo usa um princípio fundamental para assistir à detecção de linhas das pautas e evitar as dificuldades típicas de detecção de símbolos colocados em cima das linhas.

² A projecção horizontal (Y) tem como resultado um vector cujo componente i é a soma de todos os pixels pretos da linha i da imagem; a projecção vertical (X) tem como resultado um vector cujo componente j é a soma de todos os pixels pretos da coluna j na imagem.

Os resultados de detecção de símbolos musicais em partituras manuscritas estão ainda muito abaixo do desejado, pois as grandes dificuldades baseadas na variedade do tipo de escrita, na degradação do papel e da tinta, nas curvaturas das linhas, etc continuam a ser muito difíceis de ultrapassar. Para superar estas adversidades, os autores de [2] propuseram uma abordagem futura com conhecimento prévio de regras musicais no reconhecimento de símbolos.

A maneira mais usual de detectar os símbolos musicais começa por extrair símbolos musicais elementares: notas, pausas, pontos, etc. Consoante o símbolo reconhecido, este é classificado com base no tamanho, no número e na organização da secção constituinte. O reconhecimento é efectuado usando as características extraídas. A extracção é feita através de métodos como *vizinho mais próximo* durante a fase de classificação dos símbolos em [10], enquanto redes neuronais são utilizadas como classificadores em [19].

Também existem propostas de métodos estruturados, baseados em construções de gráficos para cada símbolo [11]. Estes são isolados utilizando métodos de crescimento de regiões e de adelgaçamento. Em [8] foi criado um modelo difuso (*fuzzy model*) suportado por uma detecção de símbolos robusta e "*template matching*". Este método serve para lidar com incertezas, flexibilidade e distorções ao nível do símbolo, o que é bastante importante quando se trata de símbolos manuscritos. A segmentação é efectuada em dois passos: uma análise individual dos símbolos musicais e do modelo distorcido. Na análise individual, detectam-se os segmentos verticais através de um modelo de crescimento de regiões e "*template matching*". Depois, as linhas de união (por exemplo no caso de semi-colcheias) são detectadas por um modelo de crescimento de regiões e pela transformada de *Hough*. Os símbolos restantes são extraídos mais uma vez através de "*template matching*". Deste primeiro passo resultam três hipóteses de reconhecimento; a decisão final será efectuada a partir do modelo distorcido que inclui informação do contexto musical e de regras musicais.

Na consistência gráfica, o propósito é o de alternar o grau de compatibilidade entre cada objecto e todos os objectos envolventes de acordo com as suas classes. As regras gráficas utilizadas por estes autores foram:

- Acidentes (símbolos que modificam o tom da nota que podem ser colocados no início da pauta ou atrás de notas) e a nota: a posição de um acidente está antes da nota à mesma altura;
- nota e pontos: a posição dos pontos situa-se por cima ou à frente da nota, a uma distância variável;
- entre qualquer outro par de símbolos: não se podem sobrepor.

Na consistência sintáctica, o objectivo é o de introduzir regras relacionadas com tonalidade, acidentes e métrica. Aqui, a assinatura-chave é um parâmetro relevante. O grupo de símbolos é posicionado na partitura imediatamente a seguir à clave, como uma sequência ordenada. No fim, verificam a métrica (número de batidas por compasso).

Outras técnicas para extrair e classificar símbolos musicais incluem sistemas baseados em regras para representar a informação musical, uma colecção de módulos de processamento que comunicam por uma memória comum [17] e a pesquisa de pixeis com "template matching" [12]. Baseado nas regras de escrita de música, a coerência de símbolos detectados é feita por estimação de posições sobrepostas. Também se propõe um método de reconhecimento apenas controlado por uma "gramática" que formaliza o conhecimento musical [20, 21].

2.2.4 Esquema Web Existente

A parte final num motor de construção musical é a de extrair a semântica musical de formas reconhecidas graficamente e memorizá-las numa estrutura de dados musicais. Em [5] o autor intentou os seguintes objectivos:

- Criação de uma base de dados portuguesa anotada, de pautas manuscritas de música portuguesa do século XX digitalizadas, contendo o original e a versão em *MusicXML*;
- Disponibilização de tecnologia de OMR que facilite a conversão de partituras manuscritas em notação musical standard para formato digital e a sua representação em estilo hierárquico;
- Um sistema computacional para digitalizar pautas musicais manuscritas, no formato *MusicXML*, tornando a tarefa mais rápida e menos trabalhosa;
- Motor de pesquisa *online* de pautas de música portuguesa do século XX;
- Permitir adicionar, visualizar e editar as pautas em *MusicXML* através do sistema, por via *online*;
- Aplicação web com tecnologia de reconhecimento de pautas musicais manuscritas integrando todas as partes enunciadas, criando uma solução completa, de acesso livre.

Na sua tese, alcançaram-se os objectivos. Após uma escolha cuidadosa da tecnologia a utilizar, foi desenvolvida uma base de dados e, por fim, realizada a implementação da aplicação web *OMRSYS*.

2.2.5 Resumo

Analisando o trabalho desenvolvido nesta área, conclui-se que nesta fase é necessário começar a investigar e aperfeiçoar classificadores de reconhecimento de símbolos musicais para se poder avançar. A descrição dos classificadores segue-se na secção seguinte.

2.3 Fundamentos Relevantes

Nesta secção é feita uma análise aos classificadores explicando os fundamentos relevantes inerentes a cada um deles.

2.3.1 Máquinas de Vector de Suporte (SVMs)

Máquinas de vector de suporte (SVMs), introduzidos por [22], tratam o problema de classificação como questão de otimização quadrática. Esta técnica fundamenta-se essencialmente na construção de um hiperplano como fronteira de decisão de tal modo que a margem de separação entre exemplos positivos e negativos seja maximizada. Máquinas de vectores classificam os dados utilizando vectores de suporte [23].

Sem perder a generalização, as SVMs tentam maximizar a margem do hiperplano óptimo que separa os dados. Isto é tipicamente realizado numa dimensão bastante mais elevada que a do espaço de características original. Formalmente, tendo os dados de treino $\{x_i, y_i\}_{i=1}^N$ com os dados de entrada $x_i \in \mathcal{R}^P$ e as etiquetas de classes binárias correspondentes $d_i \in \{-1, +1\}$, o hiperplano que separa linearmente de modo óptimo define-se por $g(x) = \mathbf{w}^t \boldsymbol{\varphi}(x) + b$ onde $\boldsymbol{\varphi}(x)$ denota uma transformação no espaço fixa à característica e b é o parâmetro de viés (*bias*). x é atribuído a uma classe 1 se $g(x) > 0$ ou a -1 se $g(x) < 0$. Isto é equivalente a ter $d_i[\mathbf{w}^t \boldsymbol{\varphi}(x) + b] \geq 1, i = 1, \dots, N$. Resumindo, maximizar a margem é equivalente a resolver

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2} \mathbf{w}^t \mathbf{w} \\ \text{sujeito a} \quad & d_i[\mathbf{w}^t \boldsymbol{\varphi}(\mathbf{x}) + b] \geq 1, i = 1, \dots, N \end{aligned} \quad (2.1)$$

Se as classes de treino não forem linearmente separáveis, isso quer dizer que as condições em cima registadas e a formulação do problema não podem ser mantidos. Por essa razão introduzem-se, normalmente, variáveis de folga $\xi_i, i = 1, \dots, N$. Estas permitem uma penalização para os pontos dos dados classificados erradamente. Finalmente, o objectivo é o de minimizar o erro. Isto é,

$$\begin{aligned} \min_{w,b,C,\xi_i} \quad & \frac{1}{2} \mathbf{w}^t \mathbf{w} + C \sum_{i=1}^N \xi_i \\ \text{sujeito a} \quad & d_i[\mathbf{w}^t \boldsymbol{\varphi}(\mathbf{x}) + b] \geq 1 - \xi_i, i = 1, \dots, N \\ & \xi_i \geq 0 \end{aligned} \quad (2.2)$$

onde $C > 0$ controla o compromisso (*trade-off*) entre as variáveis de desvio e a margem.

No espaço de características é mais fácil resolver o problema dual e em determinadas alturas é a única maneira de treinar as máquinas de vectores de suporte. É possível formular o problema dual para um conjunto de treino $\{x_i, y_i\}_{i=1}^N$ não separável de tal forma:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j k(x_i, x_j) \\ \text{sujeito a} \quad & \sum_{i=1}^N \alpha_i d_i = 0 \\ & 0 \leq \alpha_i \leq C \quad i = 1, 2, \dots, N \end{aligned} \quad (2.3)$$

onde $k(x_i, x_j) = \varphi^T(x_i)\varphi(x_j) = \sum_{l=0}^{m_1} \varphi_l(x_i)\varphi_l(x_j)$, $i = 1, 2, \dots, N$ e $j = 1, 2, \dots, N$. $\varphi_l(x_i)$ é o componente l na aplicação $\varphi(x_i)$ de x_i ; m_1 é a dimensão do espaço de características.

Acima descreveu-se um classificador binário enquanto que neste trabalho se enfrenta um problema de classe múltipla. De facto, são sugeridos muitos trabalhos [24], [25], [26] como extensão à classificação do problema binário que foi proposto originalmente. Geralmente existem dois tipos de abordagem para problemas de classificação de múltiplas classes. Uma é a de construir e de combinar vários classificadores binários - um-contra-um e um-contra-todos -; outra a de considerar directamente a resolução do problema quadrático. A metodologia utilizada neste trabalho foi a de um-contra-um. Este método consiste no treino das classes j e k do problema binário seguinte:

$$\begin{aligned} \min_{w^{jk}, b^{jk}, C, \xi_i^{jk}} \quad & \frac{1}{2} (\mathbf{w}^{jk})^t \mathbf{w}^{jk} + C \sum_{i=1}^N \xi_i^{jk} (\mathbf{w}^{jk})^t \\ \text{sujeito a} \quad & y_i [(\mathbf{w}^{jk})^t \boldsymbol{\varphi}(\mathbf{x}) + b] \geq 1 - \xi_i^{jk}, i = 1, \dots, N \\ & \xi_i^{jk} \geq 0 \end{aligned} \quad (2.4)$$

com l dados de treino $(x_1, y_1), \dots, (x_l, y_l)$ onde $x_i \in \mathfrak{R}^n$, $i = 1, \dots, n$ é a classe de x_i ; os dados de treino x_i são mapeados a um espaço de dimensão elevado pela função φ .

Existem três tipos comuns de máquinas de vector de suporte, dependendo do produto interno do núcleo: máquinas de aprendizagem polinomial, rede de funções de elementos radiais, também conhecida como função de núcleo de Gauss, e tangente hiperbólica. Neste trabalho utiliza-se uma função de elementos radiais, porque lida com relações de classes com atributos não lineares, tem menos parâmetros do que a polinomial e apresenta menos dificuldades numéricas:

$$k(\mathbf{x}, \mathbf{x}_i) = e^{(-\gamma \|\mathbf{x} - \mathbf{x}_i\|^2)}, \gamma \geq 0 \quad (2.5)$$

2.3.2 Redes Neurais

O termo redes neuronais foi inicialmente estudado com o objectivo de representar a informação a processar em sistemas biológicos [27], [28], [29]. As tentativas eram de produzir reconhecimento inteligente e máquinas cognitivas através da simulação da estrutura física do cérebro humano. Hoje em dia, encontram-se os princípios e algoritmos de redes neuronais em várias aplicações em diversos campos, incluindo reconhecimento de padrões e processamento de sinal.

Redes neuronais são compostas por neurónios interligados, sendo os neurónios as unidades de processamento de informação. Três elementos básicos formam o modelo neuronal: um conjunto de nós ligados (cada um multiplicado por um peso), um somador (para somar os sinais de entrada) e uma função de activação (para limitar a amplitude da saída de um neurónio) [23]. Formalmente pode-se definir a saída de um neurónio por:

$$y_k = \varphi\left(\sum_{j=1}^m \omega_{kj} x_j + b_k\right) \quad (2.6)$$

onde x_1, x_2, \dots, x_m são as camadas de entrada, w os pesos de um neurónio k , b_k é o viés e $\varphi(\cdot)$ é a função de activação. Existem três tipos comuns de funções de activação: função de limiar (*threshold*), função linear por partes e função *sigmoid*. Neste trabalho, utilizou-se a última, cujo gráfico tem forma de s. Deste modo, a função aceita valores de entrada entre -1 e +1 e retorna valores entre 0 e 1. Logo fica definido por:

$$f(x) = \frac{1}{1 + e^{(-ax)}}, \text{ onde } a > 0 \text{ representa o declive} \quad (2.7)$$

Utilizou-se uma rede neuronal com alimentação directa da entrada para a saída (*feedforward*). Tipicamente, esta rede neuronal consiste em camadas de entrada e saída de neurónios e uma ou mais camadas escondidas que não fazem parte da entrada nem da saída da rede. O sinal de entrada propaga-se pela rede do início para o fim. O algoritmo de aprendizagem utilizado para treinar esta rede foi o de *backpropagation*. Este algoritmo baseia-se na técnica de gradiente descendente, pois assim possibilita-se a minimização da função de custo. Em poucas palavras, a aprendizagem de *backpropagation error* consiste em duas fases ao longo das camadas da rede: uma passagem em frente e uma para trás. Na passagem em frente é fixada a interligação dos pesos dos neurónios. O vector de entrada é aplicado aos nós sensoriais das redes e é produzido um conjunto de saída. Na passagem para trás os pesos interligados são todos ajustados de acordo com uma regra de correlação de erro. Basicamente os valores de saída da rede são subtraídos de uma resposta (de um alvo) desejada para produzir um sinal de erro, que é depois propagado para trás pela rede para todos os neurónios. Uma rede com K saídas foi utilizada, uma correspondente a cada classe e valores de alvo de 1 para a classe correcta e 0 em outros casos.

[30] descreve no seu livro a arquitectura de um OMR. A classificação que faz dos símbolos musicais é através de redes neuronais, com resultados de desempenho entre os 80% e 90%.

2.3.3 K-Vizinho Mais Próximo (k-Nearest Neighbour)

K-vizinho mais próximo é um método de aprendizagem supervisionada para classificação de objectos baseado em exemplos de treino no espaço de características [31]. Este algoritmo pertence a um conjunto de técnicas chamadas *aprendizagem baseado na instância* (*instance-based learning*). O algoritmo é muito simples e basicamente não tem treino. Começa por estender a região local à volta de um ponto de dado x até encontrar o vizinho mais próximo k . A classe mais representada nas k amostras mais próximas define a classe prevista. Treino de dados resume-se apenas na estimativa do melhor k . Neste trabalho foi utilizado a distância euclidiana, definida por:

$$d_{euclidiana}(a, b) = \sqrt{\sum_{i=1}^d (a_i - b_i)^2}$$

2.3.4 Máquinas de Vector de Relevância (RVMs)

As SVMs não modelam a distribuição dos dados, limitam-se apenas a minimizar directamente o erro de classificação, tendo como resultado na saída uma decisão binária. Este problema pode ser atenuado com as máquinas de vector de relevância (RVMs), um classificador recente [32]. As

RVMs têm a mesma forma funcional que as SVMs, dentro de um quadro *Bayesiano*. O classificador é um modelo esparso *Bayesiano* que fornece previsões probabilísticas. A sua função de decisão depende em menos dados de entrada (isto é, é mais esparso) que as SVMs, pois estas minimizam o erro de treino com a limitação de máxima suavidade, necessitando assim mais pontos para decisão. O benefício de um classificador mais esparso é que os resultados são mais generalizáveis (reduzem o *overfitting*). As previsões das RVMs são mais confiáveis que as das SVMs pois são geradas directamente através da conclusão *Bayesiana*, enquanto as SVMs apenas podem fornecer resultados "*pseudo-probabilísticos*" através de pós-processamento (resultados entre 0 e 1). Na classificação, RVMs retornam as probabilidades de membros de cada classe em vez de estimar pontos, como é feito nas SVMs. Isto provisiona uma distribuição condicional que permite a expressão de incerteza na previsão.

As máquinas de vector de relevância foram utilizadas nesta dissertação com o objectivo de comparar um classificador desenvolvido recentemente e não utilizado nesta área com os mais comuns da literatura. De um modo geral, as máquinas de vector de suporte obtêm os melhores resultados na área de reconhecimento de padrões, mas contém desvantagens que as RVMs tentam compensar.

Com base em [32], [33] e [34] foi criado um modelo das RVMs que se adequa a este projecto de modo a se poder comparar os resultados deste novo classificador com as SVMs. As comparações são feitas apenas com as SVMs pois as características dos dois classificadores são muito semelhantes e na literatura as RVMs surgem sempre associadas às SVMs [34, 32].

2.3.4.1 O Modelo Esparso de Bayes - O Modelo Subjacente

O modelo esparso de Bayes é um tratamento *bayesiano* de um modelo linear previsível [33]. Considerando um modelo de previsão no qual existem dados de entrada $X = x_n$ com os valores de saída desejados, dados por $t = t_n, n = 1, \dots, N$, o objectivo é o de encontrar um modelo subjacente $y(x)$ que prevê t dado x e que não é influenciado por dados reais, geralmente com ruído.

Um modelo de previsão possível $y(x)$ seria um modelo linear generalizado:

$$y(x; w) = \sum_{m=1}^M w_m \phi_m(x), \quad (2.8)$$

onde $w = (w_1, w_2, \dots, w_M)$ é o vector de parâmetros ajustáveis. Devido a isto este modelo é chamado "linear nos seus parâmetros", o que conduz a vantagens analíticas. Ao mesmo tempo, se se escolher as funções base $\phi_m(x)$ de modo a serem não-lineares, $y(x; w)$ pode também ser não-linear. Se M é elevado, então este modelo é (potencialmente) muito flexível e, se a complexidade computacional e estatística do modelo for apropriadamente gerida, este pode ser aplicado de uma maneira muito eficaz a uma variedade larga de problemas.

Os RVMs são um modelo probabilístico da mesma forma funcional dos SVMs. Esparsidade é atingida através do tratamento *Bayesiano*, no qual um antecedente é introduzido sobre os pesos

governados por um conjunto que são referidos como hiperparâmetros - um destes hiperparâmetros é associado a cada peso, cujos valores mais prováveis são iterativamente estimados dos dados. A chave para a abordagem é dada pela definição deste antecedente:

$$p(w|\alpha) = \prod_{m=1}^M p(w_m|\alpha_m) \propto \prod_{m=1}^M \sqrt{\alpha_m} e^{(-\frac{\alpha_m}{2} w_m^2)} \quad (2.9)$$

onde os valores de saída desejados são dados por $t_n \in \{0, 1\}$. Para integrar a probabilidade marginal, $P(t|\alpha)$, utiliza-se analiticamente a aproximação por *Laplace*:

1. Para os valores correntes de α , os pesos mais prováveis w_{MP} são calculados utilizando a seguinte equação

$$\log\{P(t|w)p(w|\alpha)\} = \sum_{n=1}^N [t_n \log y_n + (1 - t_n) \log(1 - y_n)] - \frac{1}{2} w^T A w \quad (2.10)$$

com $y_n = \sigma\{y(x_n; w)\}$.

2. A matriz *Hessiana* w_{MP} é calculada através de:

$$\nabla\nabla \log p(t, w|\alpha)|_{w_{MP}} = -(\Phi^T B \Phi + A) \quad (2.11)$$

onde $B = \text{diag}(\beta_1, \beta_2, \dots, \beta_N)$ é uma matriz diagonal com $\beta_N = \sigma\{y(x_n)\} [1 - \sigma\{y(x_n)\}]$.

Capítulo 3

Bases de Dados e Testes Realizados

No presente capítulo descrever-se-ão as bases de dados e os testes realizados no âmbito do projecto. A análise de resultados e as respectivas conclusões serão apresentadas no capítulo seguinte.

3.1 Criação de Bases de Dados

De modo a poderem efectuar-se testes com cada um dos classificadores que foram propostos, foi necessário a criação de uma base de dados grande e variada, de modo a que um elevado número de símbolos possa ser avaliado. Para tal, é indispensável a extracção de símbolos individuais das partituras musicais disponíveis. Cada tipo de símbolo musical é depois adicionado à classe correspondente. Existem vários tipos de classes diferentes, cada uma associada a um tipo de símbolo distinto (por exemplo claves de sol, notas abertas, notas fechadas, etc.). Exemplos de classes diferentes manuscritas e sintéticas estão apresentados na figura 3.1.

Símbolos Sintéticos													
Símbolos Reais													

Figura 3.1: Diferentes tipos de classes.

Em função do tipo de teste, este número de classes pode variar, dependendo dos símbolos encontrados na base de dados das partituras musicais. Como exemplo pode referir-se a inexistência de *staccatissimi* nas partituras sintéticas.

Para o trabalho foi fornecida uma base de dados de cinco autores num formato manuscrito com um total de cinquenta partituras e uma de dezanove compositores com formato sintético com um número igual de partituras. Com base neste número, foram criadas mais 114 partituras adicionais com diversas deformações:

- 5 deformações diferentes com deformações *white speckles* para cada partitura (95 novas partituras);
- uma com deformação segundo *Kanungo* para cada partitura (19 novas partituras).

Para a criação da base de dados os símbolos foram inicialmente extraídos manualmente das partituras, o que é um trabalho exaustivo e dispendioso em tempo. Atendendo ao número elevado necessário para se efectuar um teste com resultados credíveis, cada tipo de classe deve ter um número grande de símbolos. Consequentemente o tempo investido na criação de uma base de dados para cada classe é muito demorado. Para além da extracção individual de cada símbolo, é necessário redimensioná-lo para um tamanho de 20x20, pois é o tamanho exigido por este projecto para cada classificador para cada imagem, de modo a poder realizar os testes necessários.

Neste trabalho foi utilizado o código de extracção automático de símbolos de [2]. Apesar desta detecção automática, a criação das bases de dados continua a ser muito demorada, pois tem que se proceder a uma filtragem de todos os símbolos que o algoritmo detecta por imagem, agrupá-los por classes e efectuar o seu redimensionamento. Para além disto, como o algoritmo de extracção de símbolos não apresenta robustez para graus elevados de deformações, os símbolos musicais em partituras nestas situações tiveram de ser extraídos manualmente. As seguintes base de dados foram criadas:

- Símbolos sintéticos sem deformações, 16 classes;
- símbolos reais sem deformações, 16 classes;
- combinação de símbolos de partituras reais e sintéticas, 11 classes;
- símbolos sintéticos com deformações segundo *Kanungo*, 16 classes;
- símbolos sintéticos com deformação *white speckles* (manchas brancas) ligeira, 16 classes;
- símbolos sintéticos com deformação *white speckles* acentuada, 16 classes;
- símbolos sintéticos separados por autores (19 autores no total) com o número de classes a variarem de autor para autor. Nem todas as classes estão contidas dentro das variadas partituras de cada autor;
- símbolos manuscritos separados por autores (5 autores no total) com o número de classes a variarem, dependendo do autor, visto que, tal como acontece com os símbolos sintéticos, nem todos os autores contêm nas suas partituras todas as classes de símbolos.

O número total de símbolos na soma das classes varia consoante os testes, dependendo da quantidade que foi encontrada nas partituras. De modo a obter resultados viáveis, o número de símbolos por base de dados convém ser elevado, nomeadamente acima de quinhentos, valor obtido experimentalmente após vários testes efectuados. No caso particular deste trabalho, o número é sempre acima de mil, pois também é necessário garantir que cada classe seja representada em

número suficientemente elevado, para que, na fase de treino dos classificadores, uma classe não ser apenas treinada com poucos símbolos e mais tarde, na fase de testes e validação, os símbolos serem sempre mal classificados devido a um valor baixo de atributos no treino.

Quanto mais símbolos forem utilizados, existe uma maior variedade no treino. O classificador assim aprende e ajusta melhor os parâmetros.

As deformações utilizadas¹ nas bases de dados para testar foram as seguintes:

- Deformação segundo *Kanungo* (Figura 3.2(f)), que tem como resultado uma imagem mais desfocada;
- deformação *white speckles* ligeira (Figuras 3.2(a) e 3.2(b)), que tem como efeito a adição de alguns pixels brancos à imagem;
- deformação *white speckles* acentuada (Figuras 3.2(b), 3.2(c) e 3.2(e)), que adiciona muitos pixels branco à imagem.

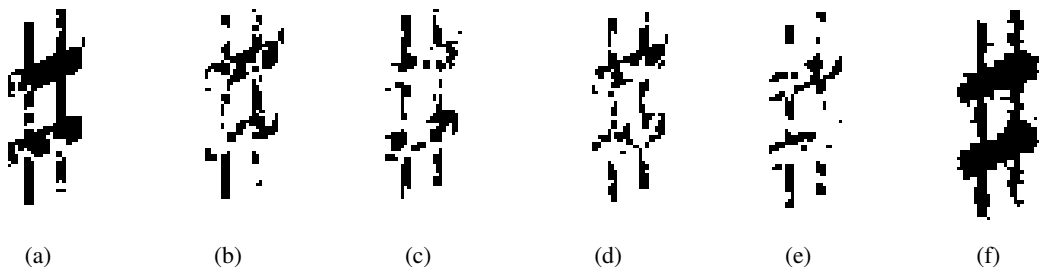


Figura 3.2: Sustenido com vários tipos de deformações: (a) *white speckles* com $w= 0.03$, (b) *white speckles* com $w= 0.05$, (c) *white speckles* com $w= 0.07$, (d) *white speckles* com $w= 0.09$, (e) *white speckles* com $w= 0.11$ e (f) deformação segundo *Kanungo* com $b = 1$ e $a = 0.5$.

3.2 Realização de Testes

Nos vários classificadores os testes realizados analisam o tamanho da base de dados por tamanho da imagem. As imagens são todas binárias, foram escaladas a um tamanho de 20×20 e, quer para treino, quer para teste e validação, são formadas matrizes da forma *número de símbolos por tamanho da imagem*. Numa fase de treino estas matrizes são analisadas coluna a coluna (neste caso de tamanho 400) e o classificador ajusta o hiperplano ou os pesos de modo às classes serem separadas da melhor forma. Estes vectores das colunas são apenas compostos por 0 e 1, dependendo se se trata de um pixel branco (0) ou de um pixel preto(1). A fase de testes serve para obter o erro na vida real. Com este erro os parâmetros de entrada no classificador são ajustados numa fase de validação.

¹Ver anexo A para mais informações sobre as deformações.

3.2.1 Especificação nos Classificadores

Os algoritmos utilizados para as experiências foram descritos na secção 2.3. Para a sua realização escolheu-se sempre uma determinada base de dados, em função do que era desejado.

A máquina de vectores de suporte tem como função de núcleo a função baseada no raio, sendo que a largura *gamma*, comum a todos os núcleos, e o valor de *c* são especificados pelo programador. Os valores de *c* encontram-se numa gama entre [2 32], enquanto *gamma* pode variar dentro do intervalo de [1/512 8]. O treino das SVMs é efectuado sobre as gamas de valores que foram atribuídos a *C* e *gamma* e no final escolhe-se o valor de acordo com o melhor desempenho. As gamas de valores que são escolhidos, tal como a função de núcleo, foram decididos consoante o mais usual na literatura [34].

De um modo análogo escolhe-se o "k" no algoritmo do k-vizinho mais próximo. Dentro de um ciclo, o classificador obtém vários resultados consoante o número de "k" existentes (neste caso entre um e cinco) e opta no final aquele que minimiza o erro.

Nas redes neuronais a escolha adequada do número de neurónios é efectuada consoante o mais usual na literatura actual. O classificador selecciona o número de neurónios que minimize o erro dentro de uma gama que vai de três a nove.

Para cada teste de cada classificador foi também sempre criada uma matriz de confusão que tem como objectivo mostrar quantos símbolos por classe são reconhecidos correctamente e quantos não. Assim, é possível no fim fazer uma análise específica em relação à dificuldade de reconhecimento de cada símbolo.

3.2.2 Separação de Autores

O objectivo destes testes é o de verificar qual o desempenho quando se está perante um cenário de treino de símbolos musicais de determinado autor e, depois, testados com outros autores. Na parte de partituras sintéticas, significa que as diferenças se podem manifestar na divergência de tipo de impressoras o que pode implicar sujidade ou manchas nos símbolos. As adversidades nas partituras manuscritas podem ser muito mais variadas. Cada autor escreve à sua maneira, o que implica à partida uma diferença da caligrafia. Também depende do tipo de utensílio utilizado para escrever nas partituras, que faz variar a grossura dos variados símbolos. Outro ponto em que se podem diferenciar é na qualidade do papel e da escrita, pois quanto mais antigo o autor, pois quanto mais antigo o autor, mais degradados se apresentam estes pontos.

3.2.3 Escolha de Características

Nesta tese foi também introduzida a ideia de efectuar uma extracção de características dos vários símbolos musicais de modo a existir mais uma técnica de reconhecer símbolos musicais.

Baseado no projecto *Gamera* [35], um *framework* para construção de aplicações de análise de documentos, foram escolhidas as características a extrair. *Gamera* não é um sistema de reconhecimento em pacotes, mas uma ferramenta para construir documentos de sistemas de reconhecimento de imagens. Facilita o desenvolvimento de novos sistemas de reconhecimento quando é aliado a algum tempo de dedicação. Visto ser um *software* livre, adequa-se bastante a este trabalho.

Directamente relacionado com esta dissertação, usa-se o formato *Gamera XML* que é utilizado para memorizar dados de imagens binárias pequenas e o mais utilizado para guardar dados de treino para um classificador. Foram utilizados os ficheiros XML criados na *framework* e passados para código *Matlab*, de maneira a serem testados e utilizados na actividade de classificação de símbolos musicais.

Considerando as características propostas em [35] foi elaborada uma análise a quinze classes de símbolos musicais para concluir que características se adequam melhor a este trabalho. Escolheram-se os símbolos de variadas partituras para se observar quais as características que se destacam melhor para uma classificação futura dos símbolos. Apenas foram levados em conta símbolos sintéticos, com o intuito de garantir resultados mais objectivos. Escolheram-se as seguintes características:

- Área de pixels pretos (normalizada): os valores obtidos correspondem à percentagem de pixels pretos do símbolo na imagem;
- orientação: retorna o ângulo de rotação da imagem;
- número de buracos (horizontal e vertical): calcula para cada linha ou coluna o valor médio de linhas brancas que não tocam nas bordas. Destes valores, a média para todas as linhas e todas as colunas é retornada;
- densidade: a razão entre volume e área dos componentes conectados. Componentes conectados com ornamentos elevados têm uma densidade baixa, enquanto um círculo perfeito tem uma densidade muito elevada;
- número de pontos finais: é aplicado um algoritmo de esqueletização à imagem e é retornado o número de pontos extremos deste esqueleto;
- intersecções: tal como acima, é aplicado um algoritmo de esqueletização à imagem e o valor retornado corresponde a intersecções do esqueleto da imagem;

Os testes efectuados com estas características têm como objectivo principal verificar o desempenho face aos classificadores quando nos confrontamos apenas com determinadas características em vez da totalidade dos símbolos. A análise é feita a uma matriz de dimensão inferior (*número de símbolos por características*) do que o referido na secção 3.2. Assim é possível diminuir o tempo de computação substancialmente.

No caso de os resultados não sejam favoráveis em termos de desempenho, será testado uma versão alternativa, que junta as características extraídas aos classificadores. Apesar de as colunas

na matriz que será analisada serem de tamanho superior às iniciais (mais sete valores) e o tempo de computação aumentar, podemos sempre obter resultados melhores que na classificação como descrita na secção [3.2](#).

Capítulo 4

Resultados e Análise

Neste capítulo, apresentam-se e analisam-se resultados obtidos ao longo do trabalho. Os vários classificadores foram testados em cenários diferentes. Mas não foram testadas todas as possibilidades para a totalidade dos classificadores. Os RVMs, que foram apenas desenvolvidos perto do final da tese, foram somente usadas para as bases de dados formadas por combinação de partituras sintéticas e manuscritas, apenas partituras sintéticas e apenas partituras manuscritas. A análise será feita por classificador e, no final, será feita uma comparação entre todos, de modo a tirar conclusões plausíveis para um trabalho futuro nesta área. Serão propostos os melhores métodos e classificadores para o reconhecimento de símbolos musicais nos diferentes cenários.

Em cada teste gerou-se também matrizes de confusão (cinco no caso das RVMs e SVMs e dez no caso das redes neuronais e no k-NN), que apresentam o número total de símbolos classificados correctamente. Não vão ser todas apresentadas aqui, mas apenas a percentagem de símbolos acertados por classe, sendo assim possível saber quais classes têm maior dificuldade de serem reconhecidas.

Os tempos de execução que inicialmente estavam pensados como alvo de análise, não serão representados nos testes, pois estes foram efectuados em vários computadores no INESC Porto, tendo cada um deles características diferentes, nomeadamente no que respeita a memória e o processamento. Assim é impossível efectuar uma análise imparcial dos tempos. Apenas é importante referir que os testes com as RVMs foram muito mais dispendiosos em tempo que qualquer outro dos classificadores, independentemente da máquina onde correu.

Inicialmente, explicar-se-á o estudo sobre a extracção de características, antes de se tecer considerações sobre a análise dos classificadores.

4.1 Extracção de Características

As características referidas na subsecção 3.2.3 foram escolhidas por serem as que melhor diferenciavam os símbolos. As restantes características propostas por [35] não se adequaram ao trabalho, visto que os resultados não eram suficientemente específicos para se diferenciar classes entre si.

Depois de obter os primeiros resultados, rapidamente se pôde concluir que não eram satisfatórios, por isso, partiu-se para a criação de um programa que fusiona o estudo da classificação dos símbolos com o estudo da extracção de características, como proposto na subsecção 3.2.3. Neste caso, é de esperar resultados iguais ou melhores do que os obtidos da simples análise aos símbolos, pois existem mais parâmetros específicos a cada classe que podem classificá-la correctamente. Para tal, alteraram-se os algoritmos de cada classificador e, na fase de testes (e na de treino e de validação também), juntaram-se às colunas das matrizes em análise mais sete valores, que correspondem ao número de características extraídas, passando assim do tamanho 400 para 407.

4.2 Resultados e Análise

Inicialmente os classificadores foram testados com uma base de dados constituída por uma combinação de partituras sintéticas e manuscritas. Passou-se, depois, para uma análise, com vários tipos de deformações, às partituras sintéticas para se verificar até que ponto ainda seria possível conseguir reconhecer símbolos. Estes testes servem, principalmente, para se simular num ambiente controlado deformações que possam existir nos símbolos manuscritos, visto a base de dados nesta área ser pequena (inicialmente estavam apenas cinquenta partituras disponíveis). Assim é possível aumentar os resultados no que diz respeito a partituras manuscritas.

Uma terceira fase de testes consistiu em verificar os resultados perante o treino de várias classes de autores específicos, sendo, depois, testado com outros autores para verificar o desempenho. Isto foi feito quer para partituras sintéticas, quer manuscritas. O objectivo é verificar a possibilidade de reconhecer símbolos de partituras, que não do mesmo autor, que tenham a mesma escrita (os resultados destes testes são principalmente interessantes no ambiente de partituras manuscritas).

Por fim, o cenário de testes enquadra-se na extracção de características, tendo inicialmente em conta apenas as características extraídas e depois a união de características e da informação do símbolo em si.

Em todas as tabelas os resultados estão apresentados como "Intervalo de Confiança (IC) para o desempenho"¹.

Também é importante referir as grandes diferenças dos tempos de execução entre as SVMs e os algoritmos de classificação das redes neuronais e do k-NN. As máquinas de vector de suporte têm um tempo de execução muito mais elevado que os outros dois classificadores. Isto deve-se ao facto de na fase de testes e treino, as SVMs demoram muito tempo a criar o modelo, isto é, carregar as imagens, efectuar a *cross-validation*, encontrar os parâmetros óptimos para *c* e *gamma* e obter o erro esperado na vida real. Com este modelo criado, a utilização deste na parte de classificação é bastante mais rápido.

¹A explicação como é obtido o intervalo de confiança é explicado em B.

4.2.1 Combinação de Partituras - Análise Apenas aos Símbolos

Classificador	Número de Símbolos	Resultados
Redes Neurais	3768	IC a 99% para o desempenho (desvio padrão): [79.68(0.65); 81.83 (2.39)] Neurónios na camada escondida = 9
k-NN	3768	IC a 99% para o desempenho (desvio padrão): [93.15(0.41); 94.58 (1.50)] K = 1
SVMs	3768	IC a 99% para o desempenho (desvio padrão): [95.41(0.07);95.94(0.25)] Valor de γ : 0.0125 Valor de C: 4.72

Tabela 4.1: Resultados dos classificadores aos testes com base de dados de combinação de partituras.

Como podemos verificar na Tabela 4.1, quer o k-Vizinho mais próximo e quer as SVMs têm resultados muito bons, com as SVMs a terem um ligeiro avanço. Já as redes neurais apresentam-se como um classificador mediano nestas condições, pois acertam apenas (aproximadamente) em oito de cada dez símbolos e com a desvantagem de demorar o dobro do tempo do que o k-NN.

Tipo de símbolo	Percentagem de detecção (em%) das redes neurais	Percentagem de detecção (em%) do k-NN	Percentagem de Detecção (em%) das SVMs
Linhas de união entre colcheias	86	98	96
Bemois	88	99	100
Bequadros	92	98	99
Notas	80	97	96
Colcheias singulares	62	90	94
Notas abertas	86	74	95
Pausas de semínimas	82	99	98
Pausas de colcheias, semi-colcheias e fusas	92	100	100
Sustenidos	88	99	98
Claves de Sol	73	93	99
Desconhecidos	50	76	81

Tabela 4.2: Percentagem de detecção dos classificadores com base de dados de combinação de partituras.

Como é de esperar, o reconhecimento dos símbolos desconhecidos é o que apresenta maior erro. Isto deve-se ao facto de os símbolos desta classe serem muito dispersos e ser muito difícil encontrar um (hiper-)plano que os diferencie de um modo ideal. Este caso irá ser igual para todas as matrizes de confusão de todos os testes.

Verifica-se na Tabela 4.2 que as SVMs obtêm sempre percentagens elevadas no que diz respeito ao número de símbolos que acertam. O k-NN apenas tem um resultado pior - nas notas abertas. Já as redes neurais demonstram-se fracas no reconhecimento de colcheias singulares e claves de sol, apresentando resultados medianos na maior parte dos restantes símbolos, somente nos bequadros e nas pausas de (semi-)colcheias evidenciam um reconhecimento acima de 90%.

4.2.2 Deformações

As deformações utilizadas nestes testes, explicadas em 3.1, podem, no caso das *white speckles*, variar. O factor de w determina o peso da deformação; quanto mais elevado o valor, maior o número de manchas brancas nas imagens. Quando se fala em *white speckles* ligeiras, o w varia entre 0.03 e 0.05. Com valores entre 0.07 e 0.11 está-se perante deformações acentuadas ².

Classificador	Número de Símbolos	Resultados
Redes Neurais	1325	IC a 99% para o desempenho (desvio padrão): [55.83(0.93); 58.92 (3.42)] Neurónios na camada escondida = 9
k-NN	1325	IC a 99% para o desempenho (desvio padrão): [82.51(0.77); 85.21 (2.83)] K = 1
SVMs	1325	IC a 99% para o desempenho (desvio padrão): [88.78(0.45); 92.33(3.79)] Valor de γ : 0.0125 Valor de C: 4.70

Tabela 4.3: Resultados dos classificadores aos testes com base de dados de símbolos sintéticos com deformações ligeiras.

Comparando os resultados da Tabela 4.3 directamente com os valores obtidos na Tabela 4.1, deparamo-nos com uma descida nos três algoritmos, sendo a descida mais acentuada visível nas redes neuronais. Isto demonstra que os símbolos com ligeiras deformações já começam a comprometer o desempenho dos classificadores. Em condições simuladas de símbolos manuscritos pode considerar-se as SVMs como o algoritmo mais adequado a ser utilizado, descartando por completa as redes neuronais, visto os resultados estarem pouco acima dos 50%.

Tipo de símbolo	Percentagem de detecção (em%) das redes neuronais	Percentagem de detecção (em%) do k-NN	Percentagem de Detecção (em%) das SVMs
Claves de dó	13	88	70
Bequadros	69	77	90
Acentos	72	93	97
Claves de fá	30	80	67
Linhas de união entre colcheias	45	71	92
Bemois	65	80	97
Notas	11	68	89
Notas Abertas	43	76	86
Colcheias singulares	74	89	96
Ligaduras	41	88	86
Pausas de semínimas	74	93	94
Pausas de colcheias, semi-colcheias e fusas	70	96	94
Sustenidos	77	87	98
Tempos quaternários e 2/2	28	88	93
Claves de sol	3	62	58
Desconhecidos	35	76	69

Tabela 4.4: Percentagem de detecção dos classificadores com base de dados de símbolos sintéticos com deformações ligeiras.

²A explicação das deformações encontra-se no Anexo A.

Como se pode observar na Tabela 4.4, as SVMs acertam, na maior parte dos casos, numa percentagem maior de símbolos do que os restantes classificadores, tal como era de esperar devido ao melhor desempenho, apresentando apenas resultados menos satisfatórios no reconhecimento de claves de sol e de claves de fá. As redes neuronais apenas apresentam nos sustentados resultados acima dos 75%, o que reforça o seu fraco desempenho nas condições de deformações leves. O k-NN apresenta uma mistura de resultados, podendo tanto alcançar bons resultados, como no caso dos acentos, como resultados menos bons, exemplo das claves de sol.

Classificador	Número de Símbolos	Resultados
Redes Neuronais	2169	IC a 99% para o desempenho (desvio padrão): [54.15(1.35); 58.63 (4.96)] Neurónios na camada escondida = 8
k-NN	2169	IC a 99% para o desempenho (desvio padrão): [63.94(0.74); 66.53 (2.72)] K = 4
SVMs	2169	IC a 99% para o desempenho (desvio padrão): [78.43(0.23); 80.23.65(0.84)] Valor de gama: 0.0125 Valor de C: 4.57

Tabela 4.5: Resultados dos classificadores aos testes com base de dados de símbolos sintéticos com deformações acentuadas.

Os resultados para deformações acentuadas, descritos na Tabela 4.5, demonstram um desempenho mais fraco dos variados algoritmos. Avaliando a prestação das redes neuronais pode desde já descartar-se este classificador como possibilidade para tentar reconhecer símbolos musicais de partituras manuscritas nas quais a qualidade de papel e da escrita esteja muito degradada. O k-NN revela um desempenho muito melhor, acertando em pouco menos de dois terços dos símbolos em análise. O único algoritmo que se aproxima de resultados aceitáveis são as máquinas de vector de suporte, daí ser aconselhável em situações de partituras manuscritas com uma grande degradação utilizar-se este método.

É ainda importante referir que, com deformações com um factor w acima de 0.10, as degradações são de tal modo elevadas, que ficam praticamente ilegíveis para um leitor humano. Considera-se, assim, que este tipo de alterações excede o aceitável para se efectuar testes que tenham interesse nesta área. Mesmo nestas condições, as SVMs acertam em cerca de 80% dos símbolos, o que pode ser considerado um excelente resultado nestas circunstâncias.

O reconhecimento por símbolo varia muito em todos os classificadores nestas condições, como se pode verificar na Tabela 4.6. Tanto podem alcançar resultados aceitáveis em determinados símbolos, como também resultados medíocres em outros. Isto justifica-se com a dispersão das classes nestas situações, que é muito elevada, o que dificulta uma separação delas por (hiper-)planos.

Os símbolos que foram testados para obter os resultados da Tabela 4.7, são os que melhor simulam símbolos patentes em partituras manuscritas. Estes não se encontram para além do seu reconhecimento óptico humano, mas apresentam pequenas falhas como são frequentes quando desenhados manualmente.

Tipo de símbolo	Percentagem de detecção (em%) das redes neuronais	Percentagem de detecção (em%) do k-NN	Percentagem de Detecção (em%) das SVMs
Acentos	35	91	87
Claves de Dó	8	56	70
Claves de fá	63	94	96
Linhas de união entre colcheias	74	63	89
Bemois	65	90	89
Bequados	44	57	71
Notas	0	23	23
Colcheias singulares	61	87	76
Notas Abertas	59	45	81
Ligaduras	12	62	
Pausas de semínimas	67	83	88
Pausas de colcheias, semi-colcheias e fusas	37	80	84
Sustenidos	69	70	89
Tempos quaternários e 2/2	0	20	0
Claves de sol	2	28	18
Desconhecidos	5	29	14

Tabela 4.6: Percentagem de detecção dos classificadores com base de dados de símbolos sintéticos com deformações acentuadas.

Classificador	Número de Símbolos	Resultados
Redes Neuronais	1642	IC a 99% para o desempenho (desvio padrão): [50.52(3.73); 63.59 (13.74)] Neurónios na camada escondida = 8
k-NN	1642	IC a 99% para o desempenho (desvio padrão): [83.81(0.85); 86.79 (3.14)] K = 1
SVMs	1642	IC a 99% para o desempenho (desvio padrão): [87.47(0.40);90.65(3.39)] Valor de γ : 0.0125 Valor de C: 4.04

Tabela 4.7: Resultados dos classificadores aos testes com base de dados de símbolos sintéticos com deformações do tipo *white speckles* ligeiras e deformações segundo *Kanungo*.

Os resultados são muito positivos, com a excepção das redes neuronais, que se distanciam muito dos outros dois classificadores a nível de desempenho. As SVMs são o classificador mais adequado a partituras manuscritas, pois apresentam os melhores resultados, rondando os 90%.

De um modo geral, na Tabela 4.8 observa-se que as SVMs tendem a ter bons resultados no reconhecimento de cada símbolo. São poucas as excepções onde a percentagem se apresenta abaixo dos 80%. Já o k-NN tem vários valores que estão abaixo daquela grandeza e as redes neuronais não têm nenhum que a exceda, justificando assim mais uma vez o seu fraco desempenho no reconhecimento de símbolos musicais.

4.2.3 Separação por Autores - Partituras Sintéticas

Para estes testes formou-se uma base de dados a partir de dezanove autores diferentes. Não é praticável arranjar para cada autor todas as classes existentes, no entanto escolheram-se catorze

Tipo de símbolo	Percentagem de detecção (em%) das redes neuronais	Percentagem de detecção (em%) do k-NN	Percentagem de Detecção (em%) das SVMs
Acentos	8	67	77
Claves de Dó	6	88	68
Claves de fá	47	84	94
Linhas de união entre colcheias	62	72	93
Bemois	67	84	91
Bequadros	65	93	91
Notas abertas	50	88	92
Notas	32	78	81
Colcheias singulares	67	88	96
Ligaduras	53	93	81
Pausas de semínimas	71	95	95
Pausas de colcheias, semi-colcheias e fusas	69	96	93
Sustenidos	76	89	96
Tempos quaternários e 2/2	31	83	95
Claves de sol	31	79	79
Desconhecidos	34	68	63

Tabela 4.8: Percentagem de detecção dos classificadores com base de dados de símbolos sintéticos com deformações do tipo *white speckles* ligeiras e deformações segundo *Kanungo*.

autores para a fase de treino e validação, que é efectuada cinco vezes, com os catorze autores a serem diferentes, acautelando-se a existência de todas as classes. Os restantes cinco autores sobram para a fase de teste, de maneira a que o classificador nunca tente validar um símbolo de uma classe desconhecida.

A base de dados neste cenário tende a ser bastante maior do em outros casos, pois tentou-se arranjar o maior número de símbolos por classe possível. Tendo em conta que são treinados sempre cinco autores, o número de símbolos em classes que tenham sempre um número elevado de ocorrências, foi limitado. Apesar desta restrição, o número continua a ser mais elevado do que nos outros testes, aumentando o tempo de execução por classificador.

Aqui não foram calculados a percentagem do número de símbolos acertados por classe, pois não é possível garantir todo o tipo de classe na fase de teste. O facto de correr cinco vezes com catorze autores, que vão variando, no treino e na validação, e cinco autores na fase de testes, que também variam, implica no final um número diferente de classes que foram testadas. O mesmo vai acontecer para a separação de autores manuscritos.

Nos resultados da Tabela 4.9 destaca-se o algoritmo do k-vizinho mais próximo, superando os resultados das SVMs. Novamente as redes neuronais obtêm o pior resultado, ficando descartadas como possível classificador nestes cenários.

Os bons resultados dos outros dois algoritmos eram de esperar, visto a variação entre tipos de impressões em partituras sintéticas ser bastante baixa. Porém, não se pode omitir o facto de os resultados serem mais fracos do que os obtidos com a combinação de partituras, o que leva à conclusão de que, por muito pequenas que sejam, as diferenças entre várias partituras sintéticas podem ser significativas. No entanto, também tem que se ter em consideração o número de símbolos por classe na parte de treino; nos testes com combinação de partituras este número era

Classificador	Número de Símbolos	Resultados
Redes Neurais	4150	IC a 99% para o desempenho (desvio padrão): [70.39(2.44); 78.52 (9.01)] Neurónios na camada escondida = 9
k-NN	4150	IC a 99% para o desempenho (desvio padrão): [90.80(0.66); 92.98 (2.42)] K = 1
SVMs	4150	IC a 99% para o desempenho (desvio padrão): [85.54(0.70); 91.10(5.93)] Valor de <i>gamma</i> : 0.0125 Valor de C: 4.81

Tabela 4.9: Resultados dos classificadores aos testes com a base de dados de partituras sintéticas separadas por autores.

elevado para qualquer classe, aqui o caso não é o mesmo. Como foi dito, nenhum autor utiliza todos os símbolos nas suas partituras e o número de cada pode ser baixo. O caso que aqui ocorre é o facto de algumas classes serem treinadas com poucos valores. Na parte de validação existe, então, uma base de dados muito inferior ao que acontece no das combinação de partituras para se poder verificar a que classe corresponde um determinado símbolo.

4.2.4 Separação por Autores - Partituras Manuscritas

Tal como em 4.2.3 foram escolhidos de autores diferentes (neste caso apenas cinco). De uma forma análoga ao realizado com as partituras sintéticas, foram sempre treinados dois autores e, posteriormente, testados os restantes três, de modo a garantir que todos os símbolos estejam presentes na fase de análise, para evitar a tentativa de validar um símbolo que seja inexistente para o classificador.

Classificador	Número de Símbolos	Resultados
Redes Neurais	3902	IC a 99% para o desempenho (desvio padrão): 33.29(2.77); 55.26 (23.46)] Neurónios na camada escondida = 7
k-NN	3902	IC a 99% para o desempenho (desvio padrão): [61.58(0.59); 66.30 (5.04)] K = 1
SVMs	3902	IC a 99% para o desempenho (desvio padrão): [55.65(1.95); 71.17(16.53)] Valor de <i>gamma</i> : 0.0125 Valor de C: 4.07

Tabela 4.10: Resultados dos classificadores aos testes com a base de dados de partituras manuscritas separadas por autores.

Como já era de esperar, os resultados apresentados na Tabela 4.10 para a base de dados de partituras manuscritas separadas por autores são bastante fracos. Tal é fácil de compreender, pois cada autor tem o seu próprio estilo de escrever que varia muito entre cada um. A dependência do utensílio de escrita também tem grande influência no resultado, visto fazer variar, dependendo da utilização de uma caneta, de um lápis ou de uma lapiseira, a grossura de cada linha no desenho de um símbolo musical (4.1).

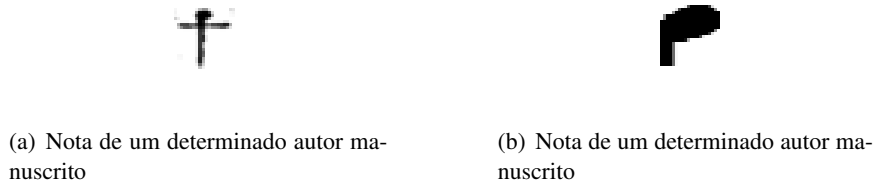


Figura 4.1: Exemplo de diferença de notas entre autores diferentes.

De notar ainda o valor elevado dos desvios-padrão (principalmente nas redes neuronais e nas SVMs). Isto demonstra uma dispersão elevada nos dados, que justifica uma vez mais os fracos resultados. As classes analisadas divergem muito em si mesmas e indicam uma grande variabilidade entre os símbolos, dificultando assim o estabelecimento das fronteiras dos conjuntos a que cada classe pertence. Com esta dificuldade acrescida torna-se também mais complicado encontrar os planos que separam melhor as classes.

4.2.5 Extração de Características

Nesta secção serão analisados os resultados obtidos pelos classificadores após a extração de características que melhor diferenciam os símbolos. Na tabela seguinte apresentam-se os resultados referentes a testes apenas com os dados que foram extraídos. A base de dados com que foi testado é a de combinação de partituras sintéticas e manuscritas.

Classificador	Número de Símbolos	Resultados
Redes Neuronais	3768	IC a 99% para o desempenho (desvio padrão): [60.55(2.07); 67.45 (7.64)] Neurónios na camada escondida = 9
k-NN	3768	IC a 99% para o desempenho (desvio padrão): [56.24(0.67); 58.40 (2.67)] K = 5
SVMs	3768	IC a 99% para o desempenho (desvio padrão): [51.49(1.40);62.50(5.14)] Valor de γ : 0.0125 Valor de C: 4.09

Tabela 4.11: Resultados dos classificadores analisando apenas as características extraídas, com a base de dados de combinação de partituras.

Os desempenhos presentes na Tabela 4.11 demonstram que o método de tentar classificar símbolos apenas através das suas características não é o adequado. Pelo facto de existirem apenas sete características que diferenciam bem os símbolos, este resultado já era de esperar. Resultados positivos teriam sido óptimos na perspectiva que a análise à base de dados ter sido feita a uma matriz de tamanho muito mais reduzido (as colunas seriam de tamanho 7 ao invés de 400 como até aqui). Mas de modo a não desperdiçar este trabalho, foi realizada uma segunda abordagem para se poder analisar os símbolos com as suas características. Dado que são mais sete pontos de diferenciação entre os símbolos, decidiu-se juntar estas sete características ao tamanho normal das

imagens em análise, ficando assim com colunas de tamanho 407 nas matrizes que são treinadas e testadas. Isto pode significar um aumento no tempo de execução, mas igualmente deve demonstrar um melhoramento no seu desempenho.

Tipo de símbolo	Percentagem de detecção (em%) das redes neuronais	Percentagem de detecção (em%) do k-NN	Percentagem de Detecção (em%) das SVMs
Linhas de união entre colcheias	82	83	85
Bemois	10	30	29
Bequadros	28	57	50
Notas	68	48	61
Colcheias singulares	14	38	22
Notas Abertas	57	64	62
Pausas de semínimas	52	57	52
Pausas de colcheias, semi-colcheias e fusas	55	55	62
Sustenidos	77	78	82
Claves de Sol	66	85	77
Desconhecidos	9	23	14

Tabela 4.12: Percentagem de detecção dos classificadores analisando apenas as características extraídas, com a base de dados de combinação de partituras.

Analisando as matrizes de confusão da Tabela 4.12, é possível tirar algumas ilações. O facto de apenas alguns símbolos obterem resultados bons (acima de 75%) justifica-se com o facto de as características que foram extraídas diferenciarem esses símbolos dos restantes da melhor maneira. Por exemplo, as linhas que unem as colcheias são, por norma, os símbolos com maior área de pixels pretos nas imagens em análise. Esta área é uma das características que foi extraída. Assim também se justifica os resultados das claves de sol, que costumam apresentar o número mais elevado de buracos na vertical.

4.2.6 Fusão de Extração de Características e de Classificadores

Classificador	Número de Símbolos	Resultados
Redes Neuronais	3768	IC a 99% para o desempenho (desvio padrão): [80.41(0.69); 82.70 (2.54)] Neurónios na camada escondida = 9
k-NN	3768	IC a 99% para o desempenho (desvio padrão): [94.04(0.1.03); 95.02 (0.28)] K = 1
SVMs	3768	IC a 99% para o desempenho (desvio padrão): [95.31(0.13);96.30(0.47)] Valor de γ : 0.0125 Valor de C: 4.89

Tabela 4.13: Resultados dos classificadores fusionando os símbolos com as características extraídas com base de dados de combinação de partituras.

Como se previa e se pode verificar na Tabela 4.13, estes resultados são, ainda que pouco, melhores que os resultados quando são analisados os símbolos por si só. Tal era de esperar pois

adicionou-se a cada símbolo mais sete pontos em análise que os diferencia melhor. Assim é mais fácil atribuir a cada classe o símbolo correctamente classificado.

Tipo de símbolo	Percentagem de detecção (em%) das redes neuronais	Percentagem de detecção (em%) do k-NN	Percentagem de Detecção (em%) das SVMs
Linhas de união entre colcheias	90	97	96
Bemois	90	99	98
Bequadros	93	99	99
Notas	83	96	97
Colcheias singulares	69	90	94
Notas Abertas	85	95	95
Pausas de semínimas	84	100	100
Pausas de colcheias, semi-colcheias e fusas	90	100	100
Sustenidos	90	98	98
Claves de Sol	79	96	96
Desconhecidos	42	70	81

Tabela 4.14: Percentagem de detecção dos classificadores fusionando os símbolos com as características extraídas com base de dados de combinação de partituras.

De acordo com os resultados dos desempenhos descritos na Tabela 4.14, a percentagem de símbolos acertados é elevada em praticamente todos. As SVMs e o k-NN apresentam resultados sempre acima dos 90% (com excepção dos símbolos desconhecidos), o que mostra que são classificadores muito homogéneos.

4.2.7 Comparação de RVMs com SVMs

No final deste trabalho foi testado o classificador mais desconhecido na literatura, as máquinas de vector de relevância. Visto aparecer na literatura como grande concorrente às máquinas de vector de suporte, as comparações são feitas directamente com este classificador. Outra razão para esta comparação directa é o facto de as SVMs obterem sempre os melhores desempenhos nos testes efectuados anteriormente.

Classificador	Número de Símbolos	Resultados
SVMs	3793	IC a 99% para o desempenho (desvio padrão): [95.41(0.07);95.94(0.25)] Valor de γ : 0.0125 Valor de C: 4.72
RVMs	3793	IC a 99% para o desempenho (desvio padrão): [90.27(0.20); 91.84 (0.73)]

Tabela 4.15: Resultados das RVMs e SVMs com base de dados de combinação de partituras.

Na Tabela 4.15 pode-se verificar o facto de o desempenho das máquinas de vector de relevância ser próximo das SVMs, acertando em mais de 90% dos símbolos.

O desempenho mais fraco das RVMs espelha-se na Tabela 4.16. O reconhecimento de símbolos como as colcheias singulares e nas notas faz baixar a percentagem geral no desempenho. Mas, de um modo geral, pode-se concluir que os resultados se assemelham com as SVMs.

Tipo de símbolo	Percentagem de detecção (em%) das RVMs	Percentagem de Detecção (em%) das SVMs
Linhas de união entre colcheias	95	96
Bemois	96	100
Bequadros	96	99
Notas	89	96
Colcheias singulares	81	94
Notas abertas	95	95
Pausas de semínima	98	98
Pausas de colcheias, semi-colcheias e fusas	100	100
Sustenidos	96	98
Claves de Sol	93	99
Desconhecidos	62	81

Tabela 4.16: Percentagem de detecção dos classificadores com base de dados de combinação de partituras.

Classificador	Número de Símbolos	Resultados
SVMs	3260	IC a 99% para o desempenho (desvio padrão): [96.75(0.21);98.42(0.78)] Valor de γ : 0.0125 Valor de C: 4.75
RVMs	3260	IC a 99% para o desempenho (desvio padrão): [89.32(0.21); 90.95 (0.76)]

Tabela 4.17: Resultados das RVMs e SVMs com base de dados de partituras manuscritas.

Tal como no caso anterior, as RVMs têm um desempenho relativamente bom, visível na Tabela 4.17. Mas a diferença de aproximadamente 7% no desempenho faz baixar a perspectiva de um uso futuro das RVMs neste cenário.

Tipo de símbolo	Percentagem de detecção (em%) das RVMs	Percentagem de Detecção (em%) das SVMs
Acentos	95	97
Claves de dó	97	98
Claves de fá	80	87
Linhas de união entre colcheias	94	97
Bemois	96	96
Bequadros	98	100
Notas	88	97
Colcheias singulares	77	87
Notas abertas	40	50
Ligaduras	73	88
Pausas de semínimas	93	97
Pausas de colcheias, semi-colcheias e fusas	98	100
Sustenidos	96	99
<i>Staccatissimi</i>	100	100
Claves de Sol	84	95
Desconhecidos	72	87

Tabela 4.18: Percentagem de detecção dos classificadores com base de dados de partituras manuscritas.

É fácil verificar através da Tabela 4.18 que em nenhum dos símbolos as RVMs obtêm um maior reconhecimento que as SVMs. Estas superam sempre os resultados, justificando a sua utilização no cenário de partituras manuscritas.

Classificador	Número de Símbolos	Resultados
SVMs	2641	IC a 99% para o desempenho (desvio padrão): [96.75(0.21);98.42(0.78)] Valor de γ : 0.0125 Valor de C: 4.73
RVMs	2641	IC a 99% para o desempenho (desvio padrão): [94.17(0.16); 95.43 (0.59)]

Tabela 4.19: Resultados das RVMs e SVMs com base de dados de partituras sintéticas.

Mais uma vez obtiveram-se resultados bastante positivos nas máquinas de vector de relevância, quase igualando o desempenho das SVMs, facto que se pode observar na Tabela 4.19. Infelizmente o seu tempo de execução é muito elevado para poder neste momento concorrer com as SVMs.

Tipo de símbolo	Percentagem de detecção (em%) das RVMs	Percentagem de Detecção (em%) das SVMs
Acentos	94	97
Claves de dó	98	96
Claves de fá	95	100
Linhas de união entre colcheias	100	98
Bemois	100	100
Bequadros	95	99
Notas	98	100
Colcheias singulares	95	99
Notas abertas	92	96
Ligaduras	72	89
Pausas de semínimas	100	100
Pausas de colcheias, semi-colcheias e fusas	92	100
Sustenidos	99	99
Tempos quaternário e 2/2	93	87
Claves de Sol	100	100
Desconhecidos	82	92

Tabela 4.20: Percentagem de detecção dos classificadores com base de dados de partituras sintéticas.

Analisando a Tabela 4.20, destaca-se que com a excepção das ligaduras, as RVMs demonstram um reconhecimento muito bom dos símbolos de cada classe, provando que neste cenário se apresenta como um bom concorrente às SVMs.

4.3 Análise Geral

Observando os vários resultados demonstrados neste capítulo, é fácil observar que o k-NN e as SVMs obtêm sempre os melhores resultados. Visto que com os testes efectuados nesta tese o modelo das SVMs já é criado, diminuindo então substancialmente o tempo de execução, pode

considerar-se este classificador como o mais robusto, visto estar quase sempre ligeiramente acima do algoritmo do k vizinho mais próximo. As redes neuronais necessitam de um melhoramento substancial se se quiser utilizá-las em trabalhos futuros.

As RVMs obtiveram também resultados positivos. Mas, como já referido, o seu tempo de execução é, por agora, muito elevado para poder concorrer com os outros classificadores.

Capítulo 5

Conclusões e Trabalho Futuro

Este capítulo destina-se a extrair as conclusões sobre o trabalho efectuado ao longo do último semestre, avaliar objectivamente os resultados e sugerir ideias para o trabalho futuro nesta área.

5.1 Conclusões

Esta dissertação centrou-se na investigação e no aperfeiçoamento de algoritmos de reconhecimento de símbolos musicais. Foi realizada uma variedade de testes em diferentes cenários, de modo a possibilitar uma análise sobre os classificadores. De um modo geral, os resultados para o reconhecimento de símbolos musicais são bons e prometedores para a continuação do trabalho neste projecto.

Pode-se concluir que, com os classificadores utilizados, obter-se símbolos musicais de determinados autores manuscritos baseados no treino de partituras de autores diferentes, leva a resultados pouco satisfatórios. A variedade de escrita entre os diversos autores é de tal maneira diferente, que dificulta a obtenção de bons resultados.

Quando se trata de classes que apresentem uma dispersão maior (geralmente isto acontece quando os símbolos dentro de uma classe têm diferenças acentuadas entre si) as SVMs têm um desempenho mais frutuoso que o dos restantes classificadores.

Em relação ao classificador menos conhecido na literatura, logo também menos utilizado em casos de classificação, os resultados em geral são bastante positivos. Em termos de desempenho podem ser considerados como bom concorrente das SVMs, apenas pecando por poucos pontos percentuais. Já o tempo de execução é um grande revés deste classificador. Infelizmente não houve tempo para se efectuar mais testes (nomeadamente testar nos outros cenários nos quais os outros classificadores também foram testados) de modo a ser possível uma conclusão mais profunda.

5.2 Trabalho Futuro

Atendendo ao trabalho efectuado neste projecto e aos resultados obtidos nesta tese de dissertação vários aspectos devem ser abordados para assegurar uma continuidade coerente.

No que diz respeito aos classificadores, um dos âmbitos de investigação deve ser o do melhoramento das RVMs. O tempo de execução é muito extenso, não sendo vantajoso em relação, por exemplo, às SVMs. De modo a poder ser feita uma comparação com os classificadores mais usuais na literatura actual, é de aconselhar mais testes com as RVMs, testando-os em cenários como o da extracção de características, o das deformações e o da separação por autores.

De modo a melhorar o funcionamento das redes neuronais, seria vantajoso estudar a sua arquitectura e alterá-la com o objectivo de melhorar o seu desempenho.

No que concerne à extracção de características pode integrar-se os classificadores com as informações obtidas para as classes da seguinte forma: sabendo que determinadas características podem identificar uma classe por si só (por exemplo, as linhas de união entre (semi-)colcheias têm sempre a maior área de pixels pretos), essas classes devem ser treinadas no classificador que apenas toma em atenção as características, sendo os restantes símbolos treinados no classificador usual (SVMs, k-NN, redes neuronais ou RVMs). Com esta integração os resultados do desempenho quando se está perante a separação por autores, podem, possivelmente, melhorar.

No caso do estudo de separação de autores de partituras em formato manuscrito, uma abordagem possível no futuro para melhorar o desempenho é basear-se em técnicas de *active learning*, em que à medida que se começa a dar ao sistema partituras de um novo autor ele vai aprendendo a reconhecer esse tipo de partituras através de algum *feedback* do utilizador.

Anexo A

Deformações

Neste anexo far-se-á uma breve descrição das deformações que foram aplicadas aos símbolos musicais nesta tese.

A.1 Deformações segundo *Kanungo*

Esta deformação tem como objectivo imitar distorções locais causadas em impressões. Este modelo tem seis parâmetros $(\eta, \alpha_0, \alpha, \beta_0, \beta, \kappa)$ [36] com o seguinte significado:

- Cada pixel preto da imagem original é invertido com a probabilidade $\alpha_0 e^{-\alpha d^2} + \eta$ onde d representa a distância ao pixel de fundo mais próximo;
- cada pixel de fundo é invertido com a probabilidade $\beta_0 e^{-\beta d^2} + \eta$, onde d representa a distância ao pixel de primeiro plano mais próximo;
- eventualmente, é executada uma operação de fecho morfológico com um disco de diâmetro κ .

A.2 Deformações *white speckles* (manchas brancas)

As deformações que geram manchas brancas (*white speckles* por entre os símbolos têm três parâmetros (p, n, k) com o seguinte significado:

- Cada pixel preto da imagem original é tomado com a probabilidade p como ponto inicial para um passo aleatório de comprimentos n ;
- uma imagem contendo o passo aleatório é suavizada através de uma operação de fecho com um rectângulo de tamanho k ;
- eventualmente, a imagem com os passos aleatórios é subtraída à imagem original resultando em manchas brancas nas posições dos passos aleatórios.

Assim, p pode ser interpretado como frequência das manchas, n como medição para as manchas e k como um factor de suavização.

Anexo B

Cálculo do Intervalo de Confiança

O cálculo do intervalo de confiança (IC) é feito da seguinte maneira:

$$\bar{X} - t^* \frac{S}{\sqrt{N}} \leq \mu \leq \bar{X} + t^* \frac{S}{\sqrt{N}} \quad (\text{B.1})$$

onde t^* é o superior $(1 - C)/2$ valor crítico para a distribuição t com $N - 1$ graus de liberdade, \bar{X} é a amostra média, S é o desvio-padrão e N é o tamanho da amostra. A variância de uma população representada por uma amostra é dada por

$$\frac{(n - 1)S^2}{\chi^2_{[1 - (\alpha/2)]}} \leq \sigma^2 \leq \frac{(n - 1)S^2}{\chi^2_{(\alpha/2)}} \quad (\text{B.2})$$

onde $\chi^2_{(\alpha/2)}$ é o valor crítico tabulado na distribuição do coeficiente chi-quadrado, abaixo do qual uma proporção de casos dados por $[1 - (\alpha/2)]$ caem.

Anexo C

Artigo Submetido a Conferência

Analysis of Classification Methods towards Musical Symbols Recognition

Andreas Seufert¹ and Ana Rebelo² and André R. S. Marçal³ and Jaime S. Cardoso⁴

Abstract.

The Optical Music Recognition (OMR) involves many research activities. However, the processing of handwritten musical scores is not yet fully incorporated in this community largely due to the poor results of the existing methods. One of the final stages in the OMR process is the recognition of musical symbols. In this paper we present an analytical study of the most common classification algorithms that can be adopted for the OMR of handwritten scores. An introduction to the Relevance Vector Machine (RVM), a probabilistic generalized linear model of identical functional form to the Support Vector Machine (SVM), is also presented. The performances of these methods are compared using both real and synthetic scores. An augmented database using four types of deformations was chosen. They induce invariances in the printed musical symbols which exist in handwritten music sheets.

The work presented here can open new research paths towards a novel automatic musical symbols recognition module for handwritten scores.

1 Introduction

Since prehistory, by Pictographs, that music is presented in our lives. Over the years, it has expanded in many several music styles and for many different purposes. All known social communities have their own musical practice. Music is a pivotal part in the cultural heritage of any society. In this way, its preservation, in all its forms, must be pursued.

The aim of OMR is the transformation of music sheets into a machine-readable symbolic format. Digitalization has been commonly used as a possible tool for preservation, offering easy duplications, distribution and digital processing. However, for operations such as search, retrieval and analysis, a recognition system is required. Unfortunately, for handwritten music the available methods are still far from providing satisfactory results. The manual process required to recognize this type of scores and to put them in relationship with the spine structure of the score is very time consuming. This justifies the research around the definition of reliable OMR algorithms for handwritten scores.

An OMR system can be composed by three principal modules: recognition of musical symbols, musical notation reconstruction and construction of a musical notation model. The first module is divided into three stages: staff line detection and removal, symbols segmentation and symbol classification. In the module of reconstruction of the

musical information, the symbols primitives are merged to form musical symbols. In the third module of final representation construction, a format of musical description is created with the information previously produced.

In this paper an analytical study of the most common classification algorithms applied to music symbols is presented. In this manner, it is focused on the step of musical symbol recognition. Our previous work [15] was a first investigation to explore these techniques. We now present a more complete analysis, with extended experimental results. A new approach based in RVM is also evaluated.

A brief review of the work done in the OMR algorithms concluded the Section 1. In Section 2 the algorithms selected for symbol classification are described. The data set adopted in the experimental evaluation is detailed in Section 3. In Section 4 the experimental results obtained with the various methods under evaluation are presented. Finally, conclusions are drawn and future work is outlined in Section 5.

1.1 State of the Art

The procedure of segmentation and recognition of musical primitives is often preceded with the detection and elimination of the staff lines. In [4] a new and robust staff line detection algorithm based on a stable path approach is presented. The proposed technique considers the staff line as the result of a global optimization problem. The image is used as a graph with staff lines connecting paths from the left margin to the right margin of the music score. As staff lines are almost the only extensive black objects on the music score, the path to look for is the shortest path between the two margins if paths (almost) entirely through black pixels are favoured.

The operation of symbol classification is in many works linked with the process of segmenting the objects from the music score [3, 5, 16, 19]. For the pattern recognition phase, many problems result from the difficulty in obtaining individual meaningful objects. This is typically due to the printing and digitalization, as well as the paper degradation over time. In addition, distortions caused by staff lines, broken and overlapping symbols, differences in sizes and shapes or zones of high density of symbols, contribute to the complexity of the operation.

One of the techniques used to isolate the musical symbols is related to the the bounding box size, the number and organization of their constituent sections [3]. Other authors [1, 10] have chosen to apply projections to detect symbols primitives. The recognition is done using features extracted from the projection profiles. In [10], the k-nearest neighbour rule is used in the classification phase, while neural networks is the classifier selected in [1].

Rossant [16] proposed a fuzzy model supported on a robust sym-

¹ INESC Porto, FEUP, email: seufert.andreas@gmail.com

² INESC Porto, FEUP, email: arebelo@inescporto.pt

³ CICGE, FCUP, email: andre.marcal@fc.up.pt

⁴ INESC Porto, FEUP, email: jaime.cardoso@inescporto.pt

bol detection and template matching for the extraction and recognition symbols. The method also incorporates graphical and syntactic rules in an effort to improve its robustness. A structural method based on the construction of graphs for each symbol is adopted by [14]. Coüasnon [5] suggested a recognition process entirely controlled by grammar, which formalizes the musical knowledge.

A more recent procedure is the reject option possibility in the k-nearest neighbor classifier [6]. However, it is not yet applied directly in the music symbols recognition. The method integrates in the classification model a confidence measure in order to reject uncertain patterns namely broken and touching symbols. Methods of this nature have gained increased attention by the scientific community [12, 17]. The advantage of this approach is the minimization of the misclassification error by choosing not to classify certain symbols.

2 Recognition Process

Towards a comparative study between classification procedures, four different approaches were evaluated: SVM, RVM, neural networks and k-nearest neighbor. Regarding the results achieved in our previous work [15] the Hidden Markov Model (HMM) was discarded due to its less satisfying performance. For a better understanding a brief explanation of the selected techniques will be provided.

Classifiers are built by taking a set of labeled examples that will be used to construct a rule that will assign a label to any new example. In other words, if we consider a general situation, we will have a training data set (x_i, y_i) ; where x_i is a feature vector for an object i , and y_i is the label associated with the object class. The relative costs of mislabeling each x_i are known and must be used to generate a decision criterion that can take any new observation vector (x_j) and assign it the correct class (y_j).

2.1 Support vector machine

One of the most widely adopted techniques by the pattern recognition community is the Support Vector Machines (SVM). This procedure has as its main idea the maximization of the separation margin between positive and negative examples having an hyperplane as the decision surface [20].

In a formal manner, given the training set $\{\mathbf{x}_i, y_i\}_{i=1}^N$ with input data $\mathbf{x}_i \in \mathbf{R}^p$ and corresponding binary class labels $d_i \in \{-1, 1\}$, the maximum-margin hyperplane is defined by $g(\mathbf{x}) = \mathbf{w}^t \varphi(\mathbf{x}) + b$ where $\varphi(\mathbf{x})$ denotes a fixed-feature space transformation and b a bias parameter; \mathbf{x} is assigned to class 1 if $g(\mathbf{x}) > 0$ or to -1 if $g(\mathbf{x}) < 0$. The maximization of the margin is equivalent to solving

$$\begin{aligned} \min_{w, b, C, \xi_i} \quad & \frac{1}{2} \mathbf{w}^t \mathbf{w} + C \sum_{i=1}^N \xi_i \\ \text{s.t.} \quad & y_i [\mathbf{w}^t \varphi(\mathbf{x}_i) + b] \geq 1 - \xi_i, i = 1, \dots, N \\ & \xi_i \geq 0 \end{aligned} \quad (1)$$

where parameter $C > 0$ controls the trade-off between the classification errors and the margin. The slack variables ξ_i , $i = 1, \dots, N$ are introduced to penalize incorrectly classified data points.

The dual formulation in (1) leads to a dependence on the data only through inner-products $\phi(\mathbf{x}_i)^t \phi(\mathbf{x}_j)$. Mercer's theorem allows us to express those inner products as a continuous, symmetric, positive semi-definite kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$ defined in the input space.

In this work, a radial-basis function network was used, given by:

$$k(\mathbf{x}, \mathbf{x}_i) = \exp(-\gamma \|\mathbf{x} - \mathbf{x}_i\|^2), \gamma \geq 0 \quad (2)$$

The binary classifier just described can be extended to multiclass scenarios. Of the multiple extensions available in the literature [13], we used the one against one methodology.

2.2 Relevance vector machine

A sparse kernel technique, similar to SVM but avoiding its principal limitations⁵, is the Relevance Vector Machine (RVM). This is based on a Bayesian approach and provides posterior probabilistic outputs [2, 18]. For a two-class classification, the method predicts the posterior probability for being one of the classes, given the input x . Formally, the logistic sigmoid function $\sigma(y) = 1/(1 + e^{-y})$ is applied to the linear model $y(x) = \sum_{n=1}^N w_n K(x, x_n) + w_0$, where $\{w_n\}$ are the model weights and $k(\cdot, \cdot)$ is a kernel function. The procedure follows a Bernoulli distribution for $P(t|x)$, so the likelihood is given by

$$P(t|w) = \prod_{n=1}^N \sigma\{y(x_n)\}^{t_n} [1 - \sigma\{y(x_n)\}]^{1-t_n} \quad (3)$$

where the targets $t_n \in \{0, 1\}$. The marginal likelihood, $P(t|\alpha)$, for the hyperparameters is analytically obtained using the Laplace approximation procedure:

1. For the current values of α , the most probable weights w_{MP} is calculated using the following equation

$$\log\{P(t|w)p(w|\alpha)\} = \sum_{n=1}^N [t_n \log y_n + (1 - t_n) \log(1 - y_n)] - \frac{1}{2} w^T A w \quad (4)$$

with $y_n = \sigma\{y(x_n; w)\}$.

2. And also the Hessian matrix at w_{MP} is computed from

$$\nabla \nabla \log p(t, w|\alpha)|_{w_{MP}} = -(\Phi^T B \Phi + A) \quad (5)$$

where $B = \text{diag}(\beta_1, \beta_2, \dots, \beta_N)$ is a diagonal matrix with $\beta_N = \sigma\{y(x_n)\} [1 - \sigma\{y(x_n)\}]$.

2.3 Neural networks

A neural network is a parametric approximation technique that has found several applications in diverse fields, including pattern recognition and signal processing. The process typically approximate a vector function f of some input x with a series of layers. Each layer forms a vector of outputs, each of which is obtained by applying the same non-linear function to different affine functions of the inputs [9].

A multi-layer perceptron (MLP), one type of a feed-forward network, was used in our evaluation. A MLP is a layered structure consisting of nodes or units (called neurons) and one-way connections or links between the nodes of successive layers. The training of the networks was carried out under Matlab[®] and was done using back-propagation together with the Levenberg-Marquardt algorithm. We used a network with K outputs, one corresponding to each class, and target values of 1 for the correct class and 0 otherwise.

⁵ The predictions of an SVM are not probabilistic, the kernel function must satisfy Mercer's condition and the error/margin trade-off C parameter needs to be estimated in the training.

2.4 k-nearest neighbor

The k-nearest neighbour algorithm is amongst the simplest of all machine learning algorithms [8]. This method belongs to a set of techniques called Instance-based Learning. It uses the heuristic that sample points near an unclassified point should indicate the class of that point. In this manner, a k -nearest neighbor classifier finds the k data points from the training set closest to the point being considered, and classifies the object with the class most common amongst its k -nearest neighbors. The training lies only in the estimation of the best k . Although the distance function could also be learnt during the training, in this work we adopted the most often used Euclidean distance.

3 Data sets

For our evaluation of the different recognition methods proposed, a data set of both real handwritten scores of 5 different composers – see Fig. 1(a) – and synthetic scores of 19 different writers – see Fig. 1(b) – was adopted. This quantitative comparison follows the one presented in [15]. The real scores consist on a set of 50 handwritten scores from 5 Portuguese musicians, with ground truth obtained manually. Images were previously binarized with the Otsu threshold algorithm [11], as implemented in the Gamera project⁶. In the synthetic data set, a number of distortions were applied – see Figure 2. This set consists on the fraction of the dataset, available from [7], written on the standard notation. For a gradual study of the classifiers performances, the deformations applied to these printed scores were curvature, rotation, Kanungo and white speckles – see [7] for more details. In total, 380 distorted images were generated from 19 perfect scores.

The relevant classes for handwritten/printed music symbols used in the training phase of the classification models is presented in Table 1. The symbols are grouped according to their shape. The rests symbols were divided into two groups – RestI and RestII. Besides that, an unknown class was included to classify those symbols that do not fit into any of the classes listed in Table 1.

In this work the different classifiers were tested for the following situations: separation of composers, gradual increase of deformations and union of real and printed scores. Each of them with a specific database of classes for each symbol⁷ that was created from the handwritten/printed music scores indicated above.

3.1 Features extraction

For this comparative study, each image of a symbol was initially resized to 20×20 pixels and then converted to a vector of 400 binary values. Besides that rather than using only the image pixels we also decided to add in the classification process 7 music symbol features. The aim was to increase the final performance of the classifier by including characteristics that distinguish similar objects. The extracted features were based on the Gamera project⁸:

1. percentage of black pixels in the 20×20 pixels window of the image;
2. orientation of the symbol;
3. the average for all rows and all columns of the average number of white runs that not touch the border;

4. ratio between volume and connected components area;
5. number of end points in the object skeleton;
6. number of intersections in the object skeleton.



(a) Handwritten scores.



(b) Synthetic scores.

Figure 1. Examples of real and synthetic scores used in this work.

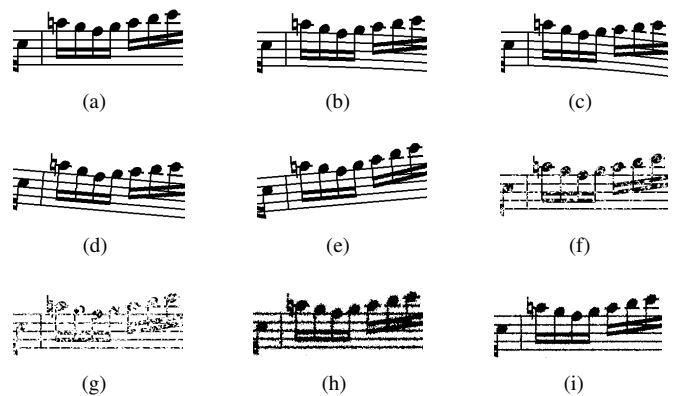


Figure 2. Examples of deformations applied to printed scores: a) Original; b) Curvature with *amplitude* = 0.04; c) Curvature with *amplitude* = 0.08; d) Rotation with *angle* = -5 ; e) Rotation with *angle* = 5; f) Noise with $p = 0.03$; g) $p = 0.07$; h) $\alpha = 0.25$; i) and $\alpha = 1.50$.

⁶ <http://gamera.sourceforge.net>

⁷ Not every class from handwritten (synthetic) symbols appear in the synthetic (handwritten) symbols, respectively.

⁸ <http://gamera.informatik.hsr.de>

Handwritten Music Symbols	>	Ⓒ	≡	♭	♮	♩	♯	⏹	⏹	⏹	♯	♩	♩
	Accent	BassClef	Beam	Flat	Natural	Note	NoteFlag	NoteOpen	RestI	RestII	Sharp	Staccatissimo	TrebleClef
Printed Music Symbols	Ⓐ	∧	≡	♭	♮	♩	♯	⏹	⏹	⏹	♯	♩	♩
	AltoClef	TieSlur	Beam	Flat	Natural	Note	NoteFlag	NoteOpen	RestI	RestII	Sharp	Time	TrebleClef

Table 1. Full set of handwritten and printed music symbols considered.

4 Results

For evaluation of the pattern recognition processes, the available dataset was randomly split into training and test sets, with 60% and 40% of the data, respectively. This division was repeated ten times in order to obtain more stable results for accuracy by averaging and also to assess the variability of this measure. No special constraint was imposed on the distribution of the categories of symbols over the training and test sets; we only guaranteed that at least one example of each category was present in the training set. The best parameterization of each model was found based on a 4-fold cross validation scheme conducted on the training set. A confidence interval was estimated for the mean of the error of the model on the test set – for more details see [15].

From the results obtained for the union of the handwritten and printed scores – see Table 2 – we can conclude that the SVMs had the best performance ([95%; 96%]), outperforming the k -NNs – the simplest model – by approximately 1% with a 99% confidence interval for the expected performance. The NNs were clearly worse than the performance of the other two classifiers. Besides that, this algorithm exhibited some difficulties with a few classes, presenting low accuracy values (NoteFlag and TrebleClef).

As expected and as it is shown in Table 3 and Table 6 the performances are slightly better than when analyzing only the raw pixels – a 400 feature vector, resulting from a 20×20 pixel image. This is due to the addition of the 7 features, so that more information to the classifier is given, which makes the recognition more robust. Looking at the classification of each class, almost every class has a better accuracy value which makes the general performance rise.

In an overall view of the results obtained with the gradual increase of deformations – see Table 4 – the achievements of the classifiers do not vary much by changing the degradation degree (except White Speckles). This leads us to conclude that they do not influence the dispersion of the classes, in other words music symbols maintain their shape. Though, the performances decay up to 30% by rising the degradation factor of the White Speckles. The main structure of the symbol gets lost with this increase. The results obtained for the degraded printed music symbols further support the superiority of the SVM model, with a 99% confidence interval for the expected performance.

From the results obtained with the separation of composers – see Table 5 – the simplest model outperformed the other, more complex, methods. On the one hand, the performance of the k -NN classifier with synthetic music sheets with a 99% confidence interval for the expected performance achieved [91%; 93%]. On the other hand, when processing handwritten musical scores, all models presented a weaker accuracy, up to 30% lower. This is due to the high variability inherent to each individual writing style.

5 Conclusions

A comparative study of classification methods for musical primitives was done in this paper. Four different pattern recognition procedures were examined namely Support Vector Machines, Neural Networks, Nearest Neighbour and Relevance Vector Machine. This study was based on two data sets of music scores containing both real handwritten and synthetic scores. Preceding the recognition phase, a detection and removal of staff lines has to be done, as well as a segmentation of the musical primitives. This operations were implemented following state of the art algorithms. A recently developed stable path approach was adopted for the staff line detection and removal.

The SVMs, RVMs, NNs and k -NNs were tested for three different situations: separation of composers, gradual increase of deformations and union of real and printed scores. The classifiers receive raw pixels – a 400 feature vector, resulting from a 20×20 pixel image – and 7 extracted characteristics as input features. These options tried to reflect standard practices in the literature. The performance of any classifier depends crucially on the choice of features. Therefore, results must be interpreted in light of these design options.

In a global overview of the results the SVMs performed best. In order to improve the results of the neural networks, a possible approach is the modification of its architecture. This idea tries to follow the proposal methods in the area of Optical Character Recognition (OCR). A potential way to enhance the performances when dealing with separation of composers is the *active learning* techniques: the OMR system learns to recognize new musical sheets through a feedback given by the user.

It is our intention, in future work, to incorporate the prior knowledge of the musical rules in the various approaches in OMR for the recognition of symbols. We expect, by doing so, to attain better results.

ACKNOWLEDGEMENTS

This work was partially funded by Fundação para a Ciência e a Tecnologia (FCT) - Portugal through project PTDC/EIA/71225/2006.

REFERENCES

- [1] P. Bellini, I. Bruno, and P. Nesi, 'Optical music sheet segmentation', *Proceedings of the 1st International Conference on Web Delivering of Music*, 183–190, (Nov. 2001).
- [2] Christopher M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [3] Dorothea Blostein and Henry S. Baird, 'A critical survey of music image analysis', in *Structured Document Image Analysis*, eds., Baird, Bunke, and Yamamoto, 405–434, Springer-Verlag, Heidelberg, (1992).
- [4] Jaime S. Cardoso, Artur Capela, Ana Rebelo, Carlos Guedes, and Joaquim Pinto da Costa, 'Staff detection with stable paths', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **31**(6), 1134–1139, (2009).

	Neural Network (%)	k -NN (%)	SVM (%)	RVM (%)
Beams	86	98	96	95
Flat	88	99	100	96
Natural	92	98	99	96
Note	80	97	96	89
NoteFlag	62	90	94	81
NoteOpen	86	74	95	95
RestI	82	99	98	98
RestII	92	100	100	100
Sharp	88	99	98	96
TrebleClef	73	93	99	93
Unknown	50	76	81	62
99% CI for the Expected performance in percentage: average (standard deviation)	[80 (0.7); 82 (2.4)]	[93(0.4); 95 (1.5)]	[95 (0.1); 96 (0.3)]	[89 (0.2); 91 (0.8)]

Table 2. Accuracy obtained with a database of real and printed scores. Total number of symbols: 3768. Input data: vector of 400 binary values.

	Neural Network (%)	k -NN (%)	SVM (%)	RVM (%)
Beams	90	97	96	94
Flat	90	99	98	97
Natural	93	99	99	98
Note	83	96	97	89
NoteFlag	69	90	94	85
NoteOpen	85	95	95	92
RestI	84	100	100	98
RestII	90	100	100	98
Sharp	90	98	98	97
TrebleClef	79	96	96	91
Unknown	42	70	81	63
99% CI for the Expected performance in percentage: average (standard deviation)	[80 (0.7); 83 (2.5)]	[94. (0.0); 95 (0.3)]	[95 (0.1); 96 (0.5)]	[90 (0.2); 92 (0.8)]

Table 3. Accuracy obtained with a database of real and printed scores. Total number of symbols: 3768. Input data: vector of 400 binary values plus 7 music symbol features.

- [5] B. Couiasnon, *Segmentation et reconnaissance de documents guidées par la connaissance a priori: application aux partitions musicales*, Ph.D. dissertation, Universit de Rennes, 1996.
- [6] Christoph Dalitz, 'Reject options and confidence measures for knn classifiers', in *Schriftenreihe des Fachbereichs Elektrotechnik und Informatik Hochschule Niederrhein*, ed., C. Dalitz, volume 8, 16–38, Shaker Verlag, (2009).
- [7] Christoph Dalitz, Michael Droettboom, Bastian Czerwinski, and Ichiro Fujigana, 'A comparative study of staff removal algorithms', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **30**, 753–766, (2008).
- [8] Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification (2nd Edition)*, Wiley-Interscience, November 2000.
- [9] David A. Forsyth and Jean Ponce, *Computer Vision: A Modern Approach*, Prentice Hall Professional Technical Reference, 2002.
- [10] Ichiro Fujinaga, 'Staff detection and removal', in *Visual Perception of Music Notation: On-Line and Off-Line Recognition*, ed., Susan George, 1–39, Idea Group Inc., (2004).
- [11] Rafael C. Gonzalez, Richard E. Woods, and Steven L. Eddins, *Digital Image processing using MATLAB*, 405–407, Upper Saddle River, NJ : Pearson/Prentice-Hall, 2004.
- [12] Yves Grandvalet, Alain Rakotomamonjy, Joseph Keshet, and Sthpane Canu, 'Support vector machines with a reject option.', in *NIPS*, eds., Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Lon Bottou, pp. 537–544. MIT Press, (2008).
- [13] Chih-Wei Hsu and Chih-Jen Lin, 'A comparison of methods for multiclass support vector machines', *Neural Networks, IEEE Transactions on*, **13**(2), 415–425, (Mar 2002).
- [14] R. Randriamahefa, J.P. Cocquerez, C. Fluhr, F. Pepin, and S. Philipp, 'Printed music recognition', *Proceedings of the Second International Conference on Document Analysis and Recognition*, 898–901, (Oct 1993).
- [15] A. Rebelo, G. Capela, and Jaime S. Cardoso, 'Optical recognition of music symbols: A comparative study', *International Journal on Document Analysis and Recognition*, (2009).
- [16] Florence Rossant and Isabelle Bloch, 'Robust and adaptive omr system including fuzzy modeling, fusion of musical rules, and possible error detection', *EURASIP Journal on Advances in Signal Processing*, **2007**(1), 160–160, (2007).
- [17] Ricardo Sousa, Beatriz Mora, and Jaime S. Cardoso, 'An ordinal data method for the classification with reject option', in *Proceedings of The Eighth International Conference on Machine Learning and Applications*, (2009).
- [18] Michael E. Tipping, 'The relevance vector machine', in *Advances in Neural Information Processing Systems 12*, eds., T. K. Leen In S. A. Solla and K.-R. Miller, 652–658, MIT Press, (2000).
- [19] Fubito Toyama, Kenji Shoji, and Juichi Miyamichi, 'Symbol recognition of printed piano scores with touching symbols', 480–483, (2006).
- [20] Vladimir N. Vapnik, *Statistical Learning Theory*, Wiley-Interscience, September 1998.

		SVM (%)	RVM (%)	Neural Network (%)	k-NN (%)
White Speckles	0.03	[89 (0.2); 91 (0.7)]	[82 (0.7); 87 (2.5)]	[80 (1.4); 85 (5.2)]	[87 (0.8); 90 (2.8)]
	0.05	[82 (0.6); 87 (2.4)]	[76 (0.7); 81 (2.4)]	[73(1.1); 77 (4.2)]	[80 (1.1); 84 (4.2)]
	0.07	[77 (0.9); 84 (3.4)]	[70 (0.6); 75 (2.3)]	[63 (2.1); 70 (7.6)]	[68 (1.4); 72 (5.1)]
	0.09	[69 (1.0); 77 (3.7)]	[61 (0.7); 67 (2.6)]	[57 (1.3); 62 (4.7)]	[56 (1.5); 61 (5.4)]
Curvature	0.02	[94 (0.6); 96 (2.2)]	[88 (0.4); 91 (1.4)]	[89 (0.9); 93 (3.4)]	[95 (0.4); 96 (1.6)]
	0.04	[92 (1.0); 95 (3.5)]	[83 (0.5); 87 (1.7)]	[87 (0.8); 90 (3.0)]	[92 (0.7); 94 (2.4)]
	0.06	[94 (0.5); 96 (2.0)]	[87 (0.4); 90 (1.6)]	[87 (1.0); 91 (3.7)]	[93 (0.8); 96 (3.1)]
	0.08	[93 (0.9); 96 (3.2)]	[83 (1.0); 91 (3.6)]	[86 (0.9); 89 (3.4)]	[91 (0.7); 94 (2.5)]
Rotation	3	[97 (0.4); 99 (1.6)]	[93 (0.3); 96 (1.1)]	[93 (0.9); 96 (3.3)]	[96 (0.7); 98 (2.5)]
	4	[98 (0.4); 100 (1.6)]	[95 (0.3); 98 (1.2)]	[95 (0.7); 97 (2.6)]	[97 (0.5); 99 (1.8)]
	5	[99 (0.2); 100 (0.6)]		[97 (0.4); 99 (1.6)]	[98 (0.4); 100 (1.5)]
	-3	[98 (0.5); 99 (1.7)]	[94 (0.3); 97 (1.2)]	[95 (0.6); 97 (2.1)]	[97 (0.6); 99 (2.1)]
	-4	[97 (0.4); 98 (1.4)]	[94 (0.4); 97 (1.6)]	[94 (0.7); 96 (2.4)]	[95 (0.4); 96 (1.3)]
	-5	[96 (0.5); 100 (1.9)]		[95 (0.6); 96 (2.0)]	[95 (0.5); 97 (2.0)]
Kanungo	0.25	[94 (0.5); 95 (1.8)]	[84 (1.5); 96 (5.5)]	[88 (1.1); 91 (4.0)]	[95 (0.4); 97 (1.6)]
	0.5	[93 (0.8); 95 (2.9)]	[87 (1.0); 95 (3.8)]	[89 (0.7); 91 (2.6)]	[95 (0.6); 97 (2.2)]
	0.75	[95 (0.8); 97 (3.1)]	[88 (0.6); 93 (2.2)]	[89 (1.0); 92 (3.5)]	[94 (0.5); 96 (1.8)]
	1	[95 (0.7); 97 (2.7)]	[89 (0.5); 93 (1.9)]	[90 (0.8); 93 (2.9)]	[94 (0.6); 96 (2.1)]
	1.25	[94 (1.0); 97 (3.6)]	[88 (0.5); 92 (1.8)]	[90 (0.8); 92 (3.0)]	[94 (0.4); 95 (1.5)]
	1.5	[95 (0.6); 97 (2.3)]	[90 (0.5); 94 (1.9)]	[91 (1.3); 95 (4.8)]	[94 (0.8); 96 (3.0)]

Table 4. Effect of different deformations on the 99% CI for the expected performance in percentage: average (standard deviation). Total number of printed symbols: 724. Input data: vector of 400 binary values.

	Total Number	SVM (%)	Neural Network (%)	k-NN (%)
Handwritten Symbols	3902	[56 (2.0); 71 (16.5)]	[50 (1.2); 59 (10.2)]	[62 (0.6); 66 (5.0)]
Printed Symbols	4150	[86 (0.7); 91 (5.9)]	[77 (1.2); 81 (4.4)]	[91 (0.7); 93 (2.4)]

Table 5. Accuracy on the 99% CI for the expected performance in percentage: average (standard deviation) for separation of composers. Input data: vector of 400 binary values.

	Total Number	SVM	
		Input Features: 400	Input Features: 407
Handwritten Symbols	3260	[95 (0.1); 97 (0.3)]	[95 (0.3); 97 (0.9)]
Printed Symbols	2641	[97 (0.3); 98 (1.1)]	[97 (0.2); 98 (0.8)]

Table 6. Accuracy on the 99% CI for the expected performance in percentage: average (standard deviation).

Referências

- [1] Ana Maria Rebelo. New methodologies towards an automatic optical recognition of hand-written musical scores. Dissertação de mestrado, Faculdade de Ciências da Universidade do Porto, October 2008.
- [2] G. Capela e Jaime S. Cardoso A. Rebelo. Optical recognition of music symbols - a comparative study. *International Journal on Document Analysis and Recognition*, (1), 2009.
- [3] Artur Capela, Jaime S. Cardoso, Ana Rebelo, and Carlos Guedes. Integrated recognition system for music scores, 2008. International Computer Music Conference (ICMC).
- [4] Artur Capela, Ana Rebelo, Jaime S. Cardoso, and Carlos Guedes. Staff line detection and removal with stable paths. In *Proceedings of the International Conference on Signal Processing and Multimedia Applications (SIGMAP 2008)*, pages 263–270, 2008.
- [5] Guilherme Artur Capela. Sistema automático de reconhecimento de pautas musicais manuscritas no inesc porto. Relatório do estágio curricular da mieic, Faculdade de Engenharia da Universidade do Porto, Setembro 2007.
- [6] Guilherme Artur Capela. Reconhecimento de símbolos musicais manuscritos na framework gamera. Dissertação do mieic, Faculdade de Engenharia da Universidade do Porto, Março 2008.
- [7] Ana Rebelo. Robust optical recognition of musical scores based on fusion of musical rules: State of the art survey, February 2009. Doctoral Programme in Electrical and Computer Engineering, Faculty of Engineering, University of Porto.
- [8] Florence Rossant and Isabelle Bloch. Robust and adaptive omr system including fuzzy modeling, fusion of musical rules, and possible error detection. *EURASIP J. Appl. Signal Process.*, 2007(1):160–160, 2007.
- [9] Jaime dos Santos Cardoso, Artur Capela, Ana Rebelo, Carlos Guedes, and Joaquim Pinto da Costa. Staff detection with stable paths. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:1134–1139, 2009.
- [10] Ichiro Fujinaga. Staff detection and removal. In Susan George, editor, *Visual Perception of Music Notation: On-Line and Off-Line Recognition*, pages 1–39. Idea Group Inc., 2004.
- [11] R. Randriamahefa, J.P. Cocquerez, C. Fluhr, F. Pepin, and S. Philipp. Printed music recognition. In *Document Analysis and Recognition, 1993., Proceedings of the Second International Conference on*, pages 898–901, Oct 1993.

- [12] Fubito Toyama, Kenji Shoji, and Juichi Miyamichi. Symbol recognition of printed piano scores with touching symbols. In *ICPR '06: Proceedings of the International Conference on Pattern Recognition*, pages 480–483, Washington, DC, USA, 2006. IEEE Computer Society.
- [13] D. Blostein and H.S. Baird. A critical survey of music image analysis. In *SDIA92*, pages 405–434, 1992.
- [14] K.T. Reed and J.R. Parker. Automatic computer recognition of printed music. *Proceedings of the 13th International Conference on Pattern Recognition*, 3:803–807 vol.3, Aug 1996.
- [15] J. V. Mahoney. Automatic analysis of music score images. B.Sc thesis, Department of Computer Science and Engineering, MIT, 1982. In Dorothea Blostein and Henry S. Baird, *A Critical Survey of Music Image Analysis*, in *Structured Document Image Analysis*, Baird, Bunke, and Yamamoto (Eds.), Eds., pp. 405–434, Springer-Verlag, Heidelberg, 1992.
- [16] D. Prerau. Computer pattern recognition of standard engraved music notation, 1970. In Dorothea Blostein and Henry S. Baird, *A Critical Survey of Music Image Analysis*, in *Structured Document Image Analysis*, Baird, Bunke, and Yamamoto (Eds.), Eds., pp. 405–434, Springer-Verlag, Heidelberg, 1992.
- [17] J. W. Roach and J. E. Tatem. Using domain knowledge in low-level visual processing to interpret handwritten music: an experiment. In Dorothea Blostein and Henry S. Baird, *A Critical Survey of Music Image Analysis*, in *Structured Document Image Analysis*, Baird, Bunke, and Yamamoto (Eds.), Eds., pp. 405–434, Springer-Verlag, Heidelberg, 1992.
- [18] I. Leplumey, J. Camillerapp, and G. Lorette. A robust detector for music staves. pages 902–905, 1993.
- [19] P. Bellini, I. Bruno, and P. Nesi. Optical music sheet segmentation. *Proceedings of the First International Conference on Web Delivering of Music*, pages 183–190, Nov. 2001.
- [20] B. Couasnon and J. Camillerapp. Using grammars to segment and recognize music scores. In *Proc. of DAS-94: International Association for Pattern Recognition Workshop on Document Analysis Systems*, pages 15–27, Kaiserslautern, 1993.
- [21] B. Couasnon, P. Brisset, and I. Stephan. Using logic programming languages for optical music recognition. *International Conference on the Practical Application of Prolog*, pages 115–34, 1995.
- [22] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.
- [23] Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999.
- [24] Vojtěch Franc and Václav Hlaváč. Multi-class support vector machine. Technical report, 2002.
- [25] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002.
- [26] E. Mayoraz and E. Alpaydim. Support vector machines for multiclass classification. Technical report, 1998.
- [27] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. pages 15–27, 1988.

- [28] Bernard Widrow and Marcian E. Hoff. Adaptive switching circuits. pages 123–134, 1988.
- [29] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. pages 673–695, 1988.
- [30] Paolo Nesi Kia Ng. *Interactive Multimedia Music Technologies (Premier Reference Source Series)*. Idea Group Inc (IGI), Secaucus, NJ, USA, 2008.
- [31] Keinosuke Fukunaga. *Introduction to statistical pattern recognition (2nd ed.)*. Academic Press Professional, Inc., San Diego, CA, USA, 1990.
- [32] Michael E. Tipping. Sparse bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.*, 1:211–244, 2001.
- [33] Michael E. Tipping. An efficient matlab implementation of the sparse bayesian modelling algorithm (version 2.0), 2009. This document is intended as a basic user-guide and implementation overview for the SparseBayes Version 2.0 software package for Matlab,.
- [34] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [35] Christoph Dalitz Ichiro Fujinaga, Michael Droettboom. The gamera framework. Disponível em <http://gamera.informatik.hsnr.de>, acessado a última vez em 06 de Janeiro de 2010.
- [36] Tapas Kanungo, Robert M. Haralick, Henry S. Baird, Werner Stuezel, and David Madigan. A statistical, nonparametric methodology for document degradation model validation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1209–1223, 2000.