

SCOREREADER – PROTÓTIPO PARA RECONSTRUÇÃO DE PARTITURAS, ATRAVÉS DE ELEMENTOS IDENTIFICADOS A PARTIR DE IMAGENS

Aluno: Alan Soares Carneiro

Orientador: Marcel Hugo

Roteiro da apresentação

- Introdução
- Objetivos
- Fundamentação Teórica (Pilares)
- Correlatos (Estado da arte)
- O Interpress (Protótipo estendido)
- Especificação
- Resultados
- Dificuldades / Conclusões
- Sugestões

Introdução

- Interesse por música (Pesquisa / Execução);
- Software de apoio ao músico;
- Extensão Interpress;
- Acessibilidade;
- Preservação da cultura musical.

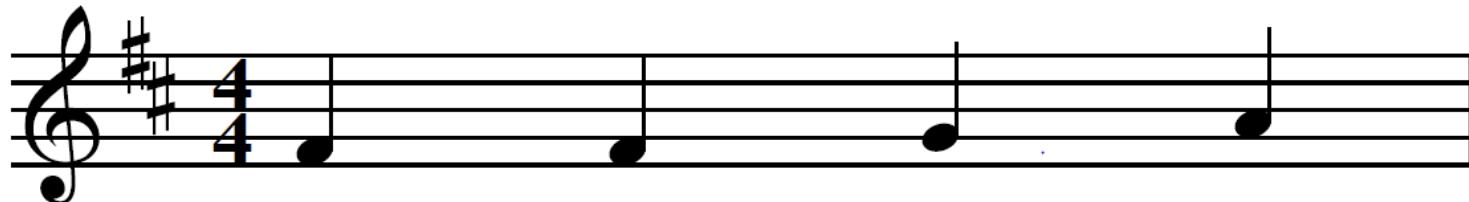
Objetivos

- Torná-lo um OMR – Optical Music Recognition;
- Melhoria etapa de remoção das linhas da pauta (Musicstaves);
- Melhoria etapa de classificação (Aprendizado por maquina);
- Reconstrução da partitura (Notação ABC);
- Saídas em formatos Notação ABC, PDF, MIDI e MUSICXML (Acessibilidade pessoas com deficiência visual);

Fundamentação Teórica (Conceitos, Técnicas e Ferramentas utilizadas)

Pauta (Notação Musical)

Onde a musica é escrita



Símbolos que representam notas ou sons e suas respectivas durações



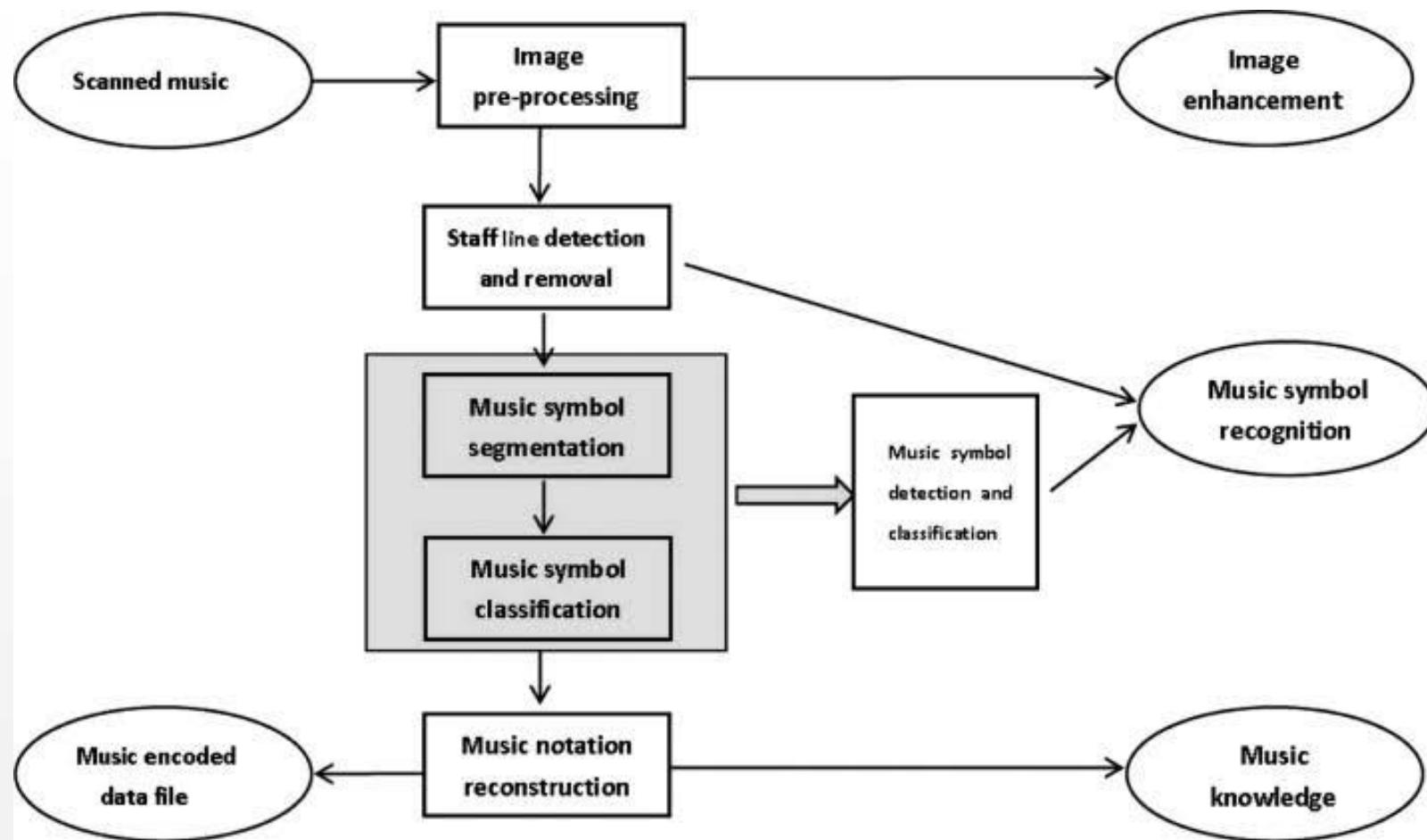
Demais símbolos



OMR – Optical Music Recognition

Em vez de confiar na intuição, é preferível defini-lo como sendo o campo de pesquisa que investiga como ler notação musical em documentos de forma computacional (CALVO-ZARAGOZA; HAJIĆ JUNIOR; PACHA, 2019).

Etapas de um OMR

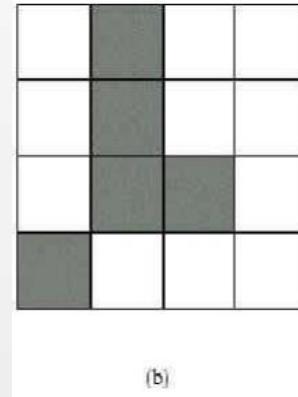
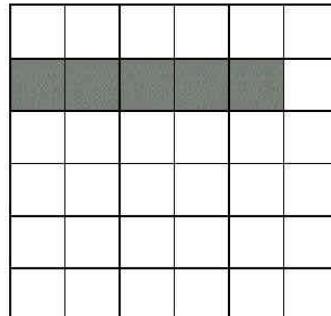
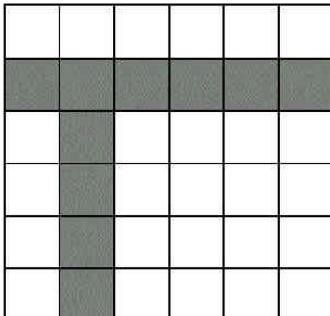


PDI - Processamento Digital de Imagens

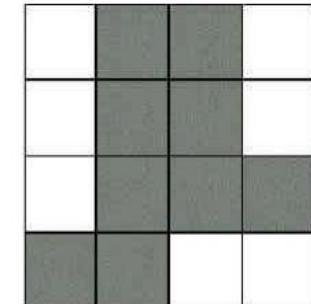
Inicia com a captura de uma imagem, que corresponde à iluminação que é refletida na superfície de objetos. A imagem passa por um processo chamado de digitalização, onde é representada de forma apropriada para o tratamento computacional (Queiroz; Gomes, 2006)

Morfologia matemática operações elementares

Dilate - Engordar ou conectar grãos próximos Erode – Diminuir partículas



(b)



(c)

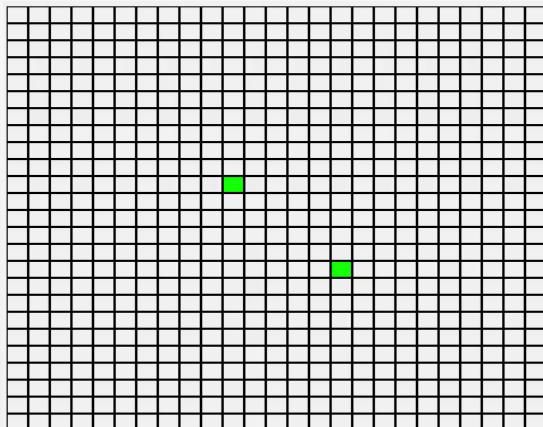
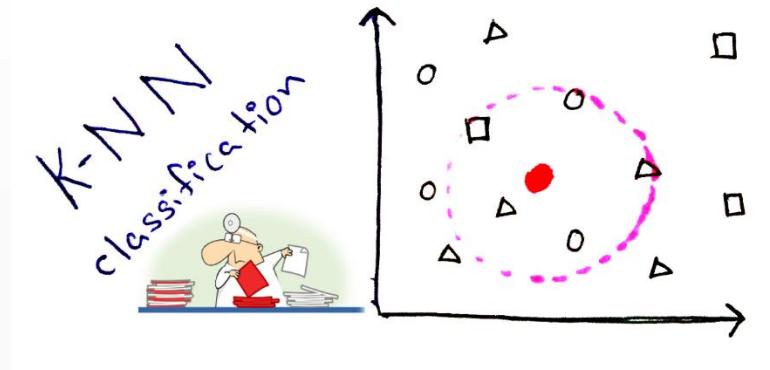
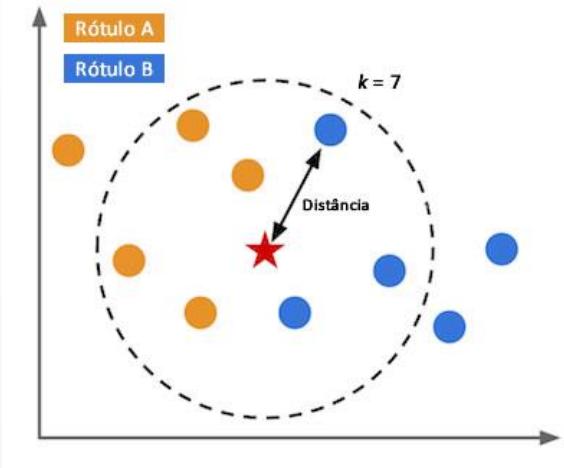
Machine Learning

Aprendizado por máquina

O termo aprendizado por máquina ou Machine Learning, refere-se ao funcionamento de sistemas computacionais capazes de aprender ou ter seu comportamento modificado através de experiências acumuladas durante sua operação (STANGE;2011).

KNN – k-nearest neighbor (Scikit-learn)

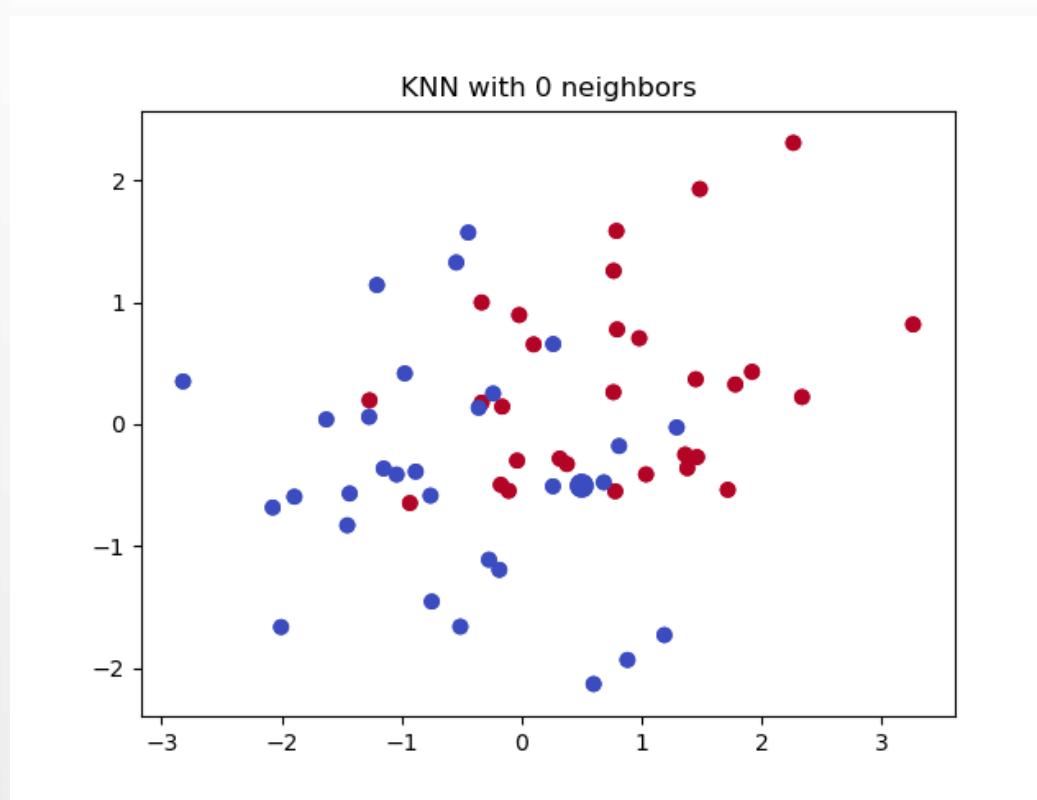
Baseado em distâncias



- Hipótese que dados similares tendem estar concentrados na mesma região no mesmo espaço de DISPERSAO
- Algoritmo preguiçoso (Lazzy)
- Considera proximidade para realizar as predições
- Menor distância classificado com rotulo da classe correspondente

KNN – k-nearest neighbor (Scikit-learn)

- Calcular distância
- Ordenar do menor para maior [distância – classe correspondente]
- K é o ponto de corte (k deve ser ímpar)



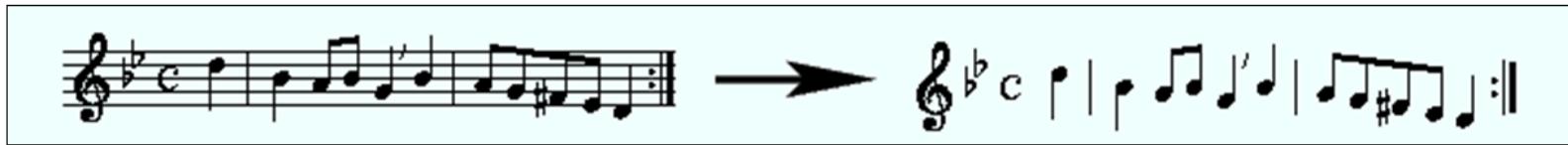
Distancia	Classe
1	C1
2	C2
3	C2
4	C1
5	C1

K = 2

K = 5

MUSICSTAVES (Python API)

Também trabalho acadêmico



Fonte: Dalitz et al. (2008, p. 1).

Two staves of handwritten musical notation. The top staff is in common time (indicated by 'C') and has a treble clef. It features various note heads, stems, and rests, with some strokes indicating grace notes or specific performance techniques. The bottom staff is also in common time ('C') and has a treble clef. It contains similar musical elements, including a dynamic marking 'tr' (trill) over a note. Both staves show a variety of note values and rests, typical of complex musical scores.

Notação ABC (Linguagem)

Utiliza-se de caracteres ASC como forma de representar notação musical computacionalmente

Notes



X:1
T:Notes
M:C
L:1/4
K:C
C, D, E, F, | G, A, B, C | D E F G | A B c d | e f g a | b c' d' e' | f' g' a' b' |]

Fonte: Walshaw (1995).

Compiladores (Notação ABC)

	Nome	Data de modificaç...	Tipo	Tamanho
1	abc2midi.exe	08/09/2019 23:46	Aplicativo	156 KB
2	abc2xml.exe	01/12/2019 16:13	Aplicativo	3.700 KB
3	abcm2ps.exe	08/09/2019 23:46	Aplicativo	455 KB
	midi.bat	09/09/2019 03:01	Arquivo em Lotes ...	1 KB
	musicxml.bat	09/09/2019 02:27	Arquivo em Lotes ...	1 KB
	svg.bat	09/09/2019 02:27	Arquivo em Lotes ...	1 KB

Fonte: elaborado pelo autor.

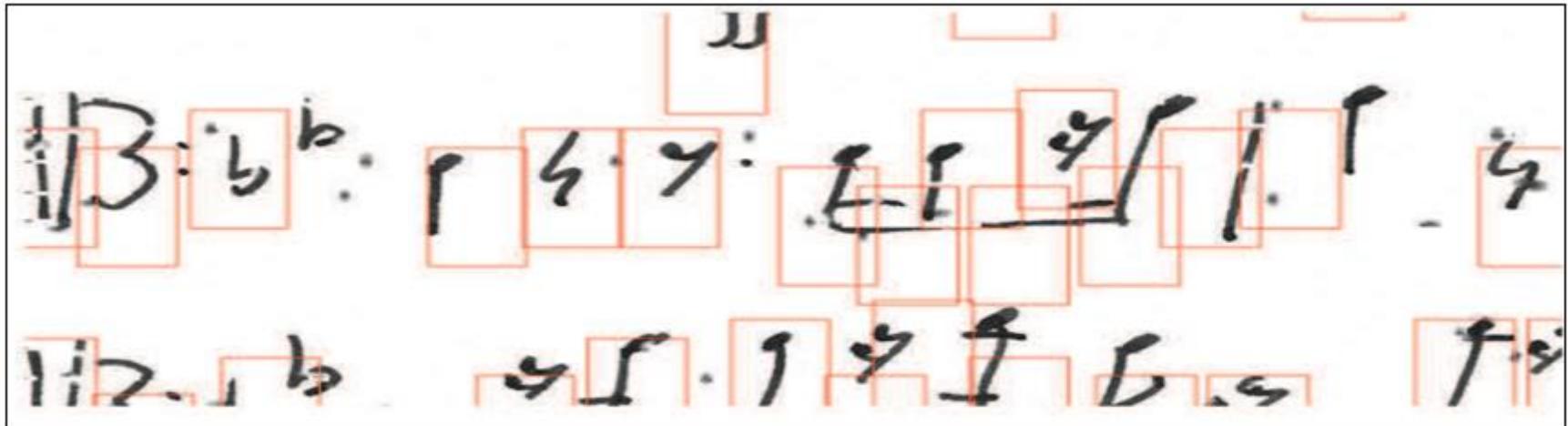
Identificador	Descrição
1	Compilador de código em notação ABC para MIDI
2	Compilador de código em notação ABC para MUSICXML
3	Compilador de código em notação ABC para SVG

Fonte: elaborado pelo autor.

Correlatos (Estado da Arte)

Castro (2014)

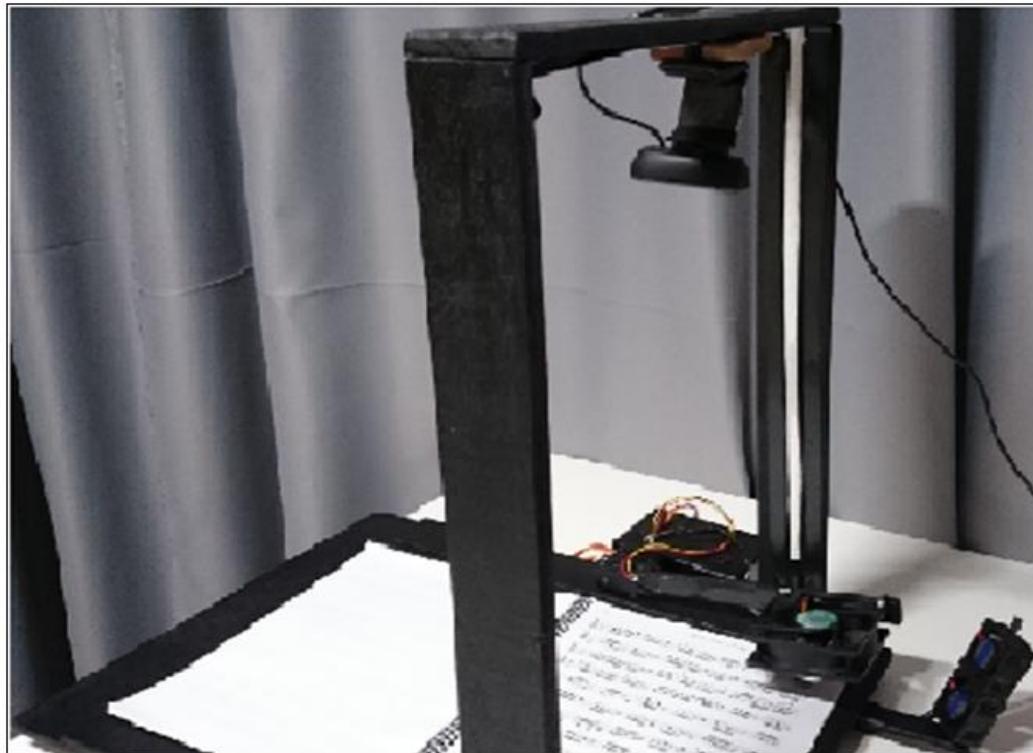
- Reconhecimento de símbolos em partituras manuscritas em imagens em tom cinza.



Fonte: Castro (2014, p. 37).

Jesus (2015)

- Protótipo autônomo (hardware e software) copista;
- Saída formato MIDI.



Fonte: Jesus (2015, p. 9).

Silva (2017)

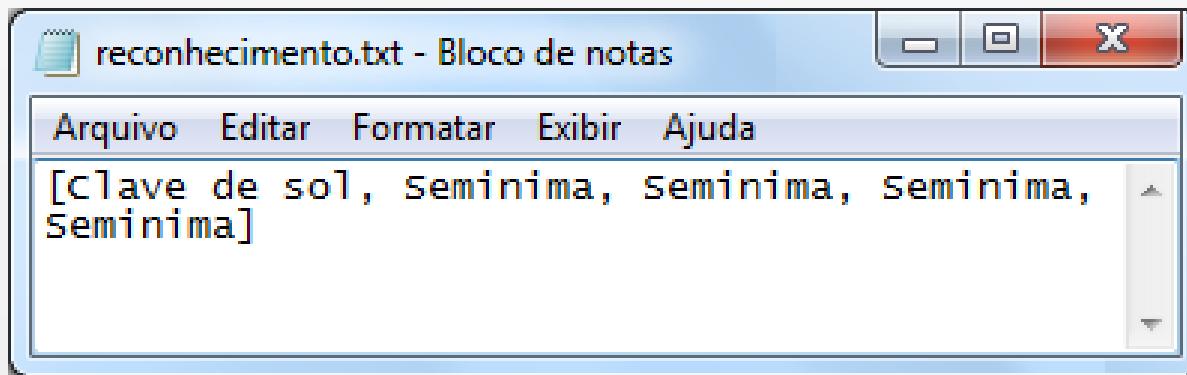
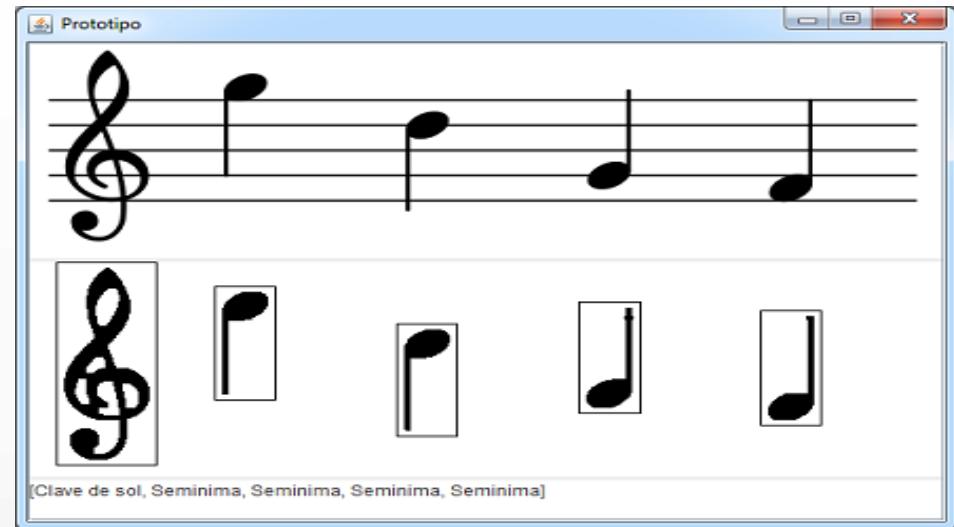
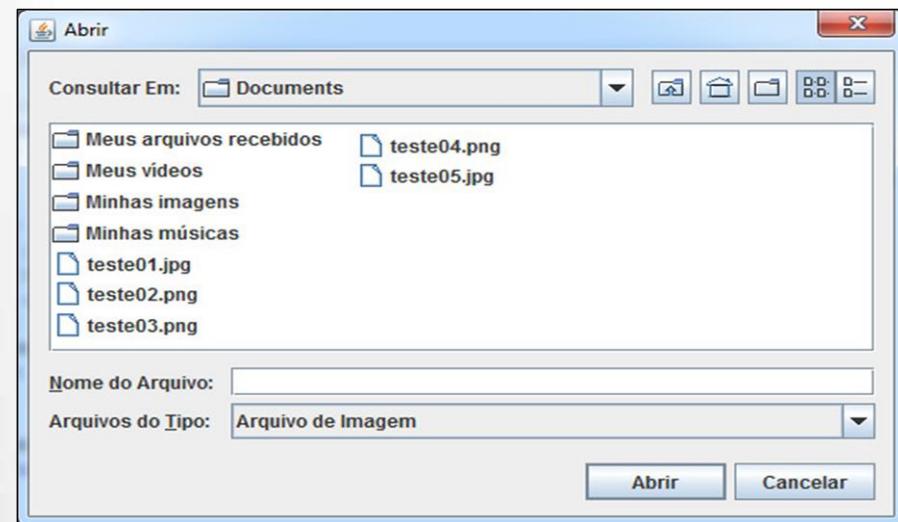
- Reconhecimento de partituras manuscritas;
- Etapas completas de um OMR.



Fonte: Silva (2017, p. 53).

O Interpress Mauricenz (2013)

Telas do Interpress



Especificação

Requisitos Funcionais

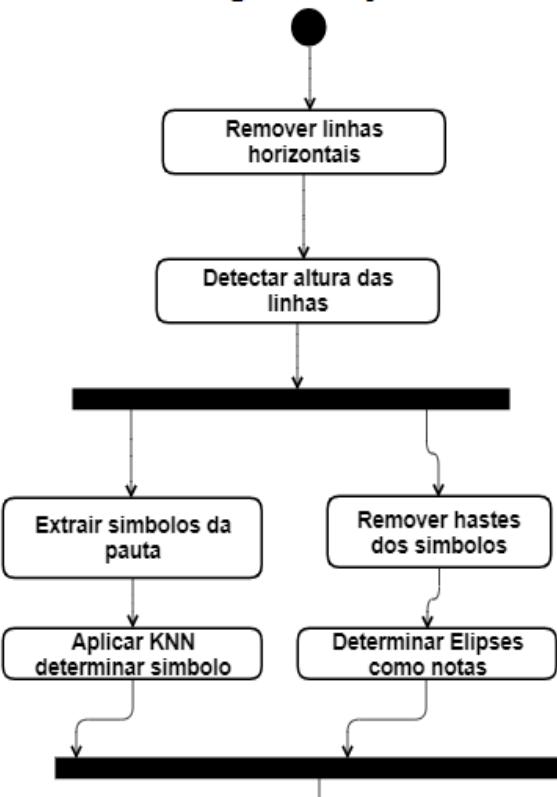
- Pré-processar e segmentar imagem;
- Reconstruir partitura em notação ABC;
- Apresentar passo a passo etapas do processamento da imagem de partitura;
- Permitir rotular símbolos e efetuar treinamento do classificador no protótipo;
- Gerar saída em formato PDF;
- Gerar saída em formato MIDI;
- Gerar saída em formato MUSICXML.

Requisitos Não Funcionais

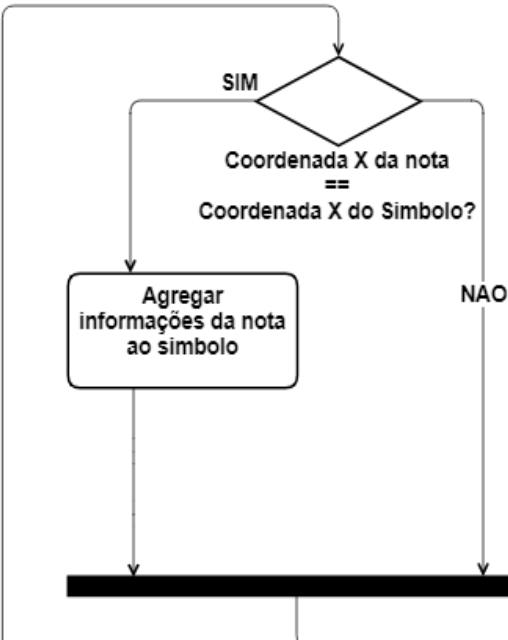
- Ser desenvolvido em arquitetura cliente servidor;
- Utilizar linguagem de programação Java no Front-End;
- Utilizar linguagem de programação Python no Back-End;
- Utilizar biblioteca do MUSICSTAVES para remoção das linhadas da pauta;
- Utilizar biblioteca OpenCV para segmentação das imagens;
- Utilizar algoritmo KNN da biblioteca scikit-learn para treinamento e classificação dos símbolos.

Diagrama atividades

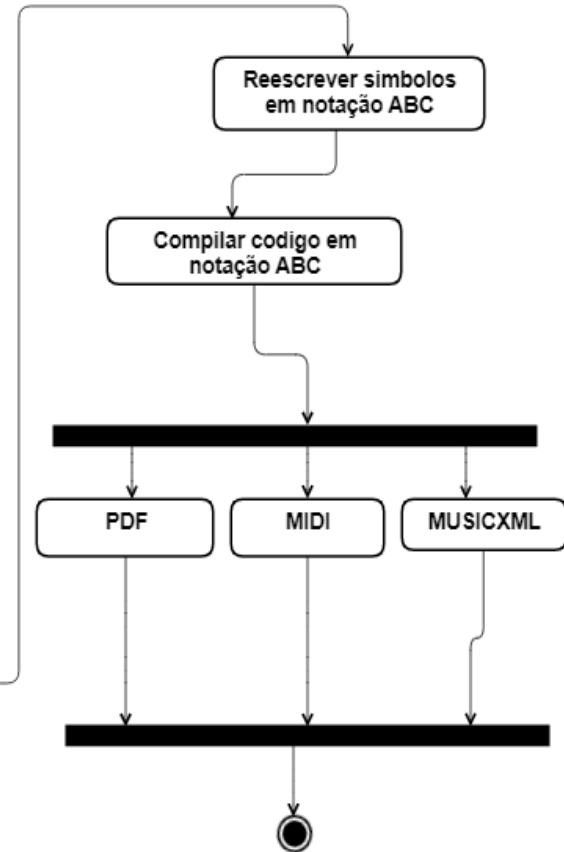
Segmentação



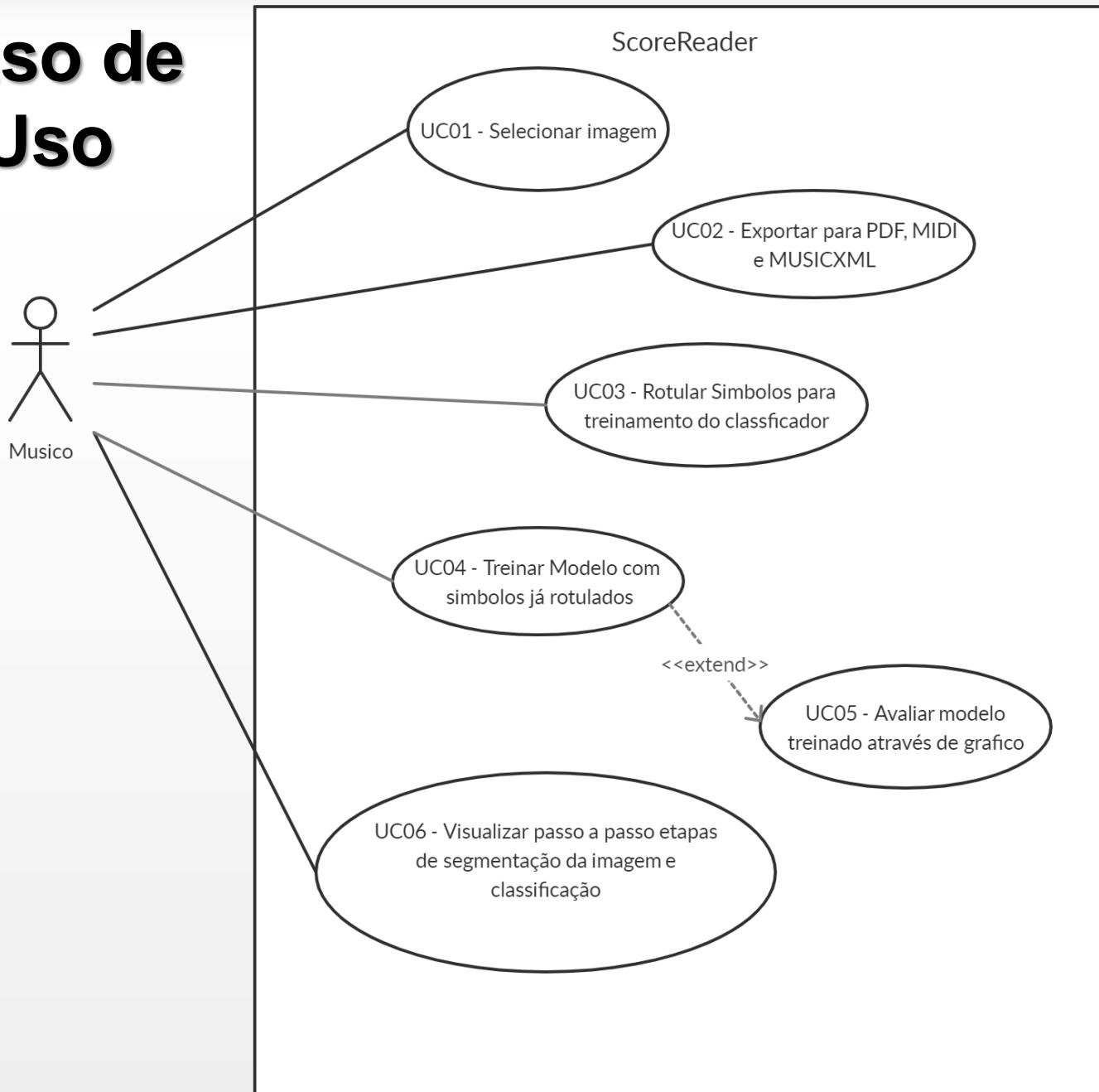
Classificação



Reconstrução

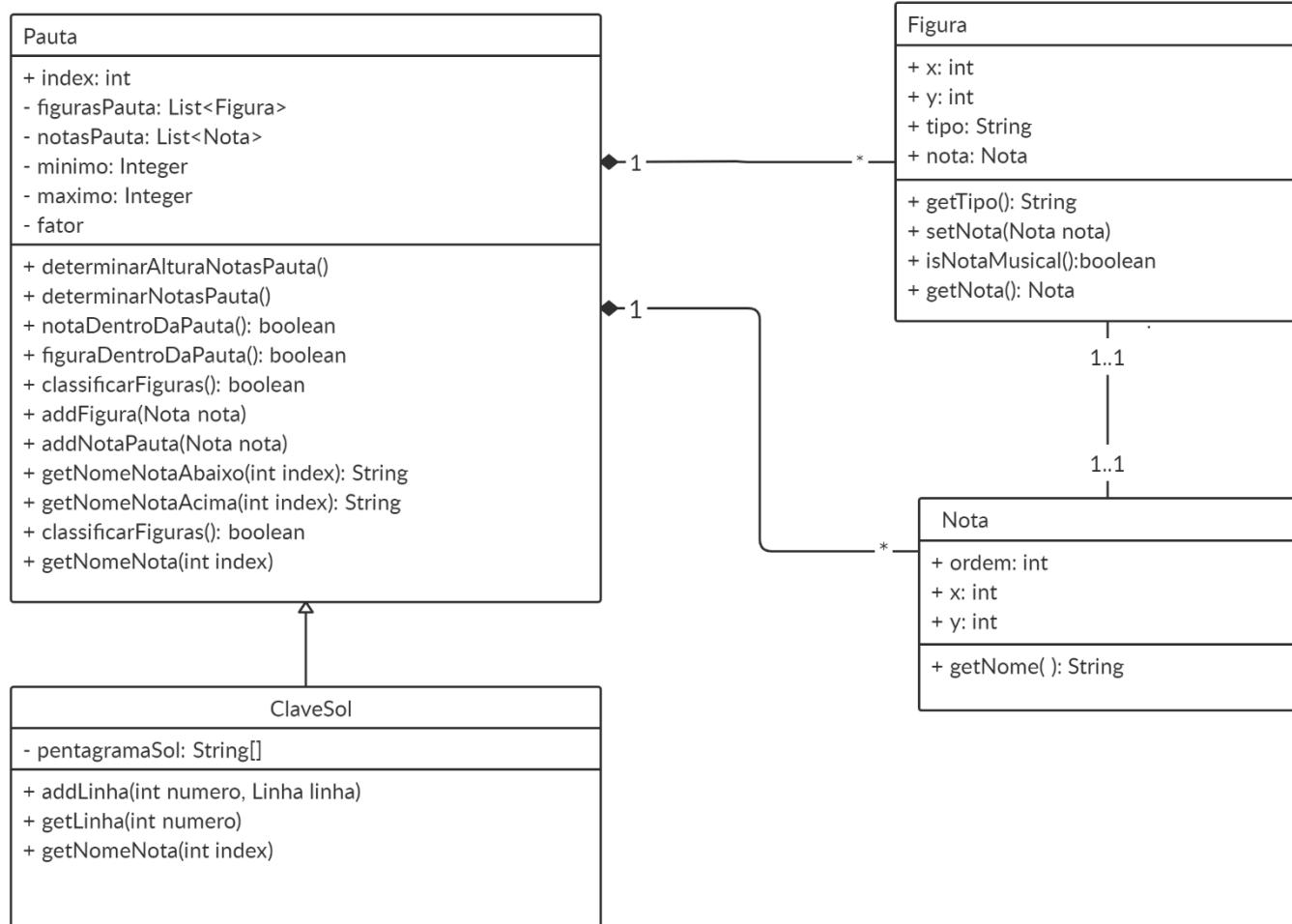


Caso de Uso

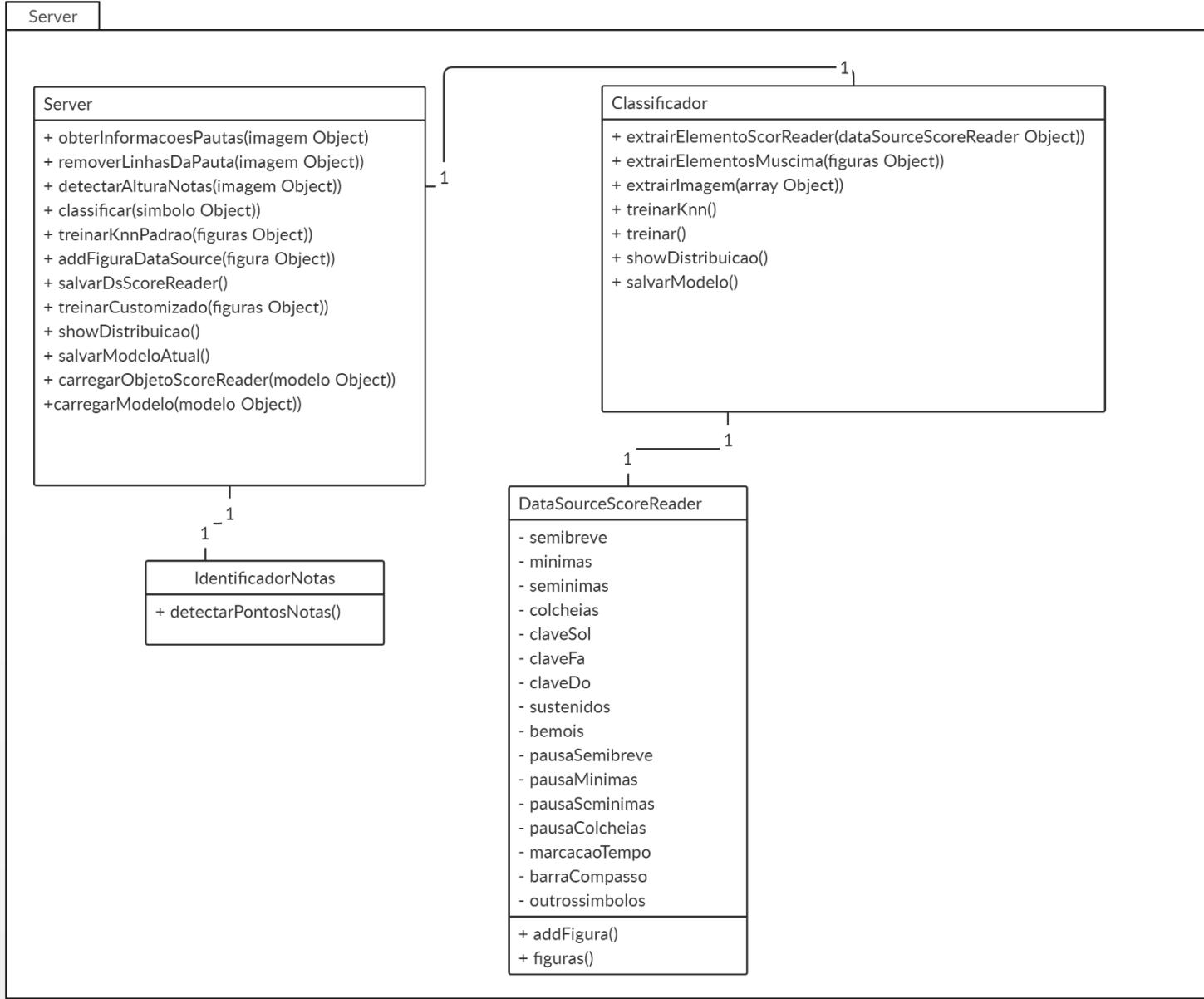


Classes (Java) – Front-End

ScoreReader



Classes (Python) – Back-End



Implementação

- Arquitetura Cliente e Servidor;
- Java como linguagem de Front-End;
- Python como linguagem de Back-End;
- Reaproveitamos e modificamos algoritmos do Interpress;
- Externas Java;
 - Compiladores Notação ABC (PDF, MIDI, MUSICXML)
 - OpenCV (Pré-processamento e Segmentação)
 - BATIK (PDF)
- Externas Python
 - MUSICSTAVES (Remover linhas da pauta)
 - SCIKIT-LEARN (KNN – IA)
 - OpenCV (Detecção de formas, Segmentação)

O Protótipo

Tela principal

Score Reader 1.0.0

Image Etapas Treinamento

1 Score Reader 1.0.0

Servidor: http://localhost:8090/

Modelo: KNN_PADRAO

Tipo: DIGITAL

Debug: Files Imagens

Arquivo: irmeiro\Desktop\TCC\ScoreReader\repository\Twinkle Twinkle Little Star.png

Abrir Editor Imagens Rotular

Processar Conteúdo Partitura Compilar

Formatos: ABC SVG XHTML

MusicXML PDF MIDI

Imagem original

2

3

```
--- Carregando modelo KNN_PADRAO
--- Modelo carregado
--- Partitura sendo processada, por gentileza aguarde, isso pode levar um certo tempo...
--- Partitura processada com sucesso...
--- Index: 0 Figura: ClaveSol
--- Index: 1 Figura: Sustenido
--- Index: 2 Figura: Sustenido
--- Index: 3 Figura: MarcacaoTempo
--- Index: 4 Nota: D Figura: Seminima
--- Index: 5 Nota: D Figura: Seminima
--- Index: 6 Nota: A Figura: Seminima
--- Index: 7 Nota: A Figura: Seminima
--- Index: 8 Figura: BarraCompasso
--- Index: 9 Nota: B Figura: Seminima
--- Index: 10 Nota: B Figura: Seminima
--- Index: 11 Nota: A Figura: Minima
--- Index: 12 Figura: BarraCompasso
```

Rotulador

Sel 1.0.0

Pausa Semibreve Minima Seminima Sustenidos Clave Sol Clave Fa Outros Agrupar Remover

Colcheia Semi Colcheia Barra Compasso Bemois Clave Do Tempo Enviar

Selecionando: Seminima 10

Rotular

11

Semibreve Minimas Semimas Colcheias Barra Compasso Sustenidos Bemois

12

Semibreve

Minimas

Semimas

Colcheias

Barra Compasso

Sustenidos

Bemois

Etapas

Score Reader 1.0.0

Image Etapas Treinamento

1) Remover Linhas da pauta - Detectar elementos musicais contidos na imagem

cm 0 cm 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37

4

7

8

9

10

11

12

2) Segmentar elementos musicais encontradas

cm 0 cm 1 2 3 4 5 6 7

cm 0 cm 1 2 3 4 5 6 7

cm 0 cm 1 2 3 4 5 6 7

cm 0 cm 1 2 3 4 5 6 7

cm 0 cm 1 2 3 4 5 6 7

5

1

2

3

4

5

6

7

Etapas

Score Reader 1.0.0

Image Etapas Treinamento

3) Detectar altura das linhas da pauta

cm 0 cm 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
0 cm
1
2
3
4
5
6
7
8
9

3) Segmentar possíveis figuras encontradas

cm 0 cm 1 2 3 4
0 cm
1
2
3
4

cm 0 cm 1 2 3 4
0 cm
1
2
3
4

cm 0 cm 1 2 3 4
0 cm
1
2
3
4

cm 0 cm 1 2 3 4
0 cm
1
2
3
4

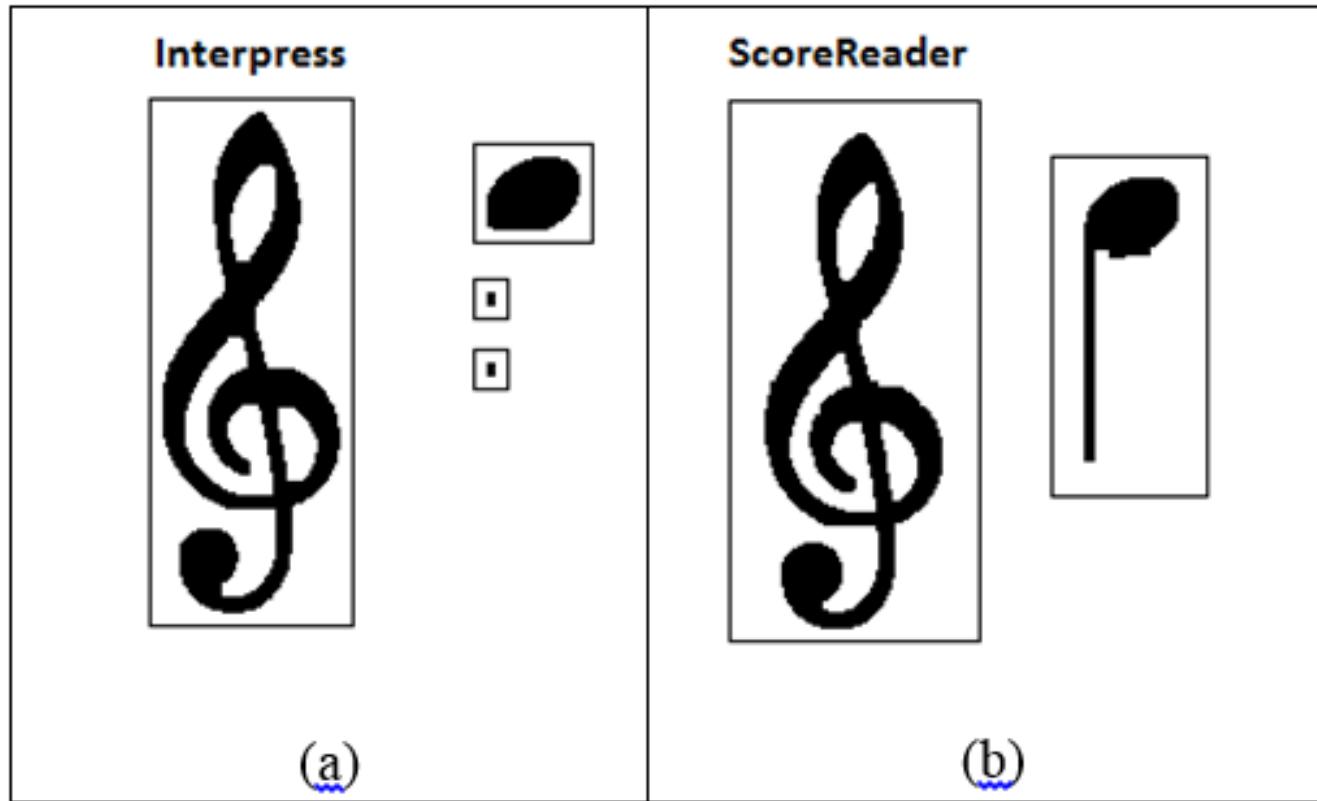
Etapas

Treinamento



Resultados

Pré-processamento



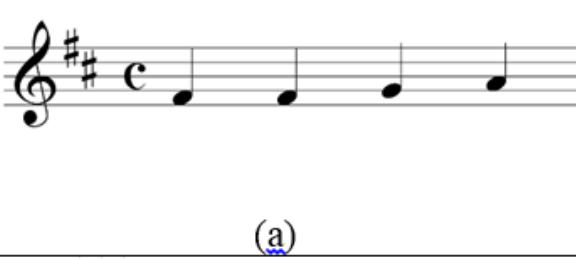
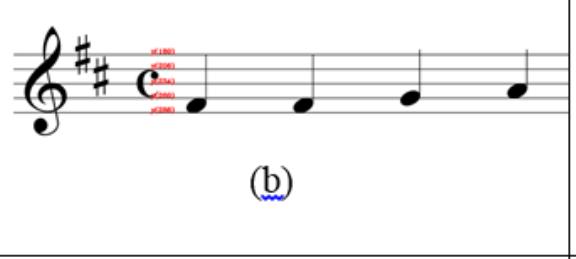
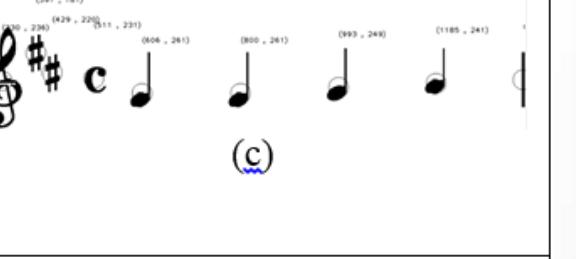
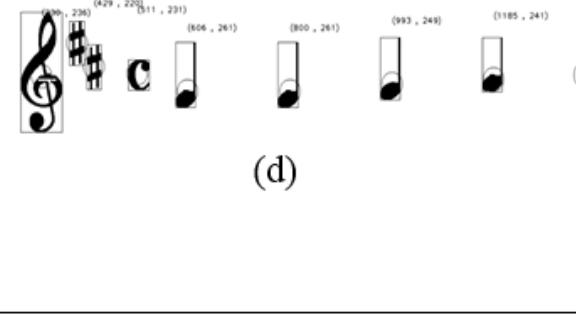
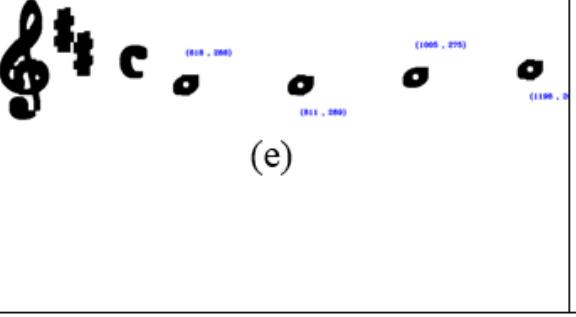
Experimentos iniciais do protótipo Estendido

Objetivo de garantir se realmente houve alguma evolução no reconhecimento

Testes	Símbolos	ScoreReader (2019)	Interpress (2013)	
Experimento 1	5	5	5	
Experimento 2	8	8	8	
Experimento 3	4	4	-	
Experimento 4	20	20	17	
	37	37	30	

Fonte: elaborado pelo autor.

Resultado (Etapas)

 (a)	 (b)	 (c)
 (d)	 (e)	<p>1 X: 1Exemplo ScoreReader 2 T: Exemplo ScoreReader 3 C: Transcrito por prototip 4 K:D treble 5 M:4/4 6 F2F2G2A2]</p> (f)
		(g)

Fonte: Elaborado pelo autor.

Resultado (Escala musical)

Score Reader 1.0.0

Image Etapas Treinamento

Score Reader 1.0.0

Servidor: http://localhost:8090/

Modelo: carneiro\Desktop\TCC\ScoreReader\treinamento\DIGITAL_KNN_PADRAO.pkl

Debug: Files Imagens

Arquivo: top\TCC\ScoreReader\repository\Casos\Musicas\Beethoven 2 compassos.png

Abrir Editor Imagens Rotular

Processar Conteúdo Partitura Compilar

Formatos: ABC SVG XHTML

Musical PDF MIDI

Carregando Modelo C:\Users\lascarneiro\Desktop\TCC\ScoreReader\treinamento...

Modelo carregado

Partitura sendo processada, por gentileza aguarde, isso pode levar um co...

Partitura processada com sucesso...

Index: 0 Figura: ClaveSol

Index: 1 Figura: Sustenido

Index: 2 Figura: Sustenido

Index: 3 Figura: MarcacaoTempo

Index: 4 Nota: F Figura: Seminima

Index: 5 Nota: F Figura: Seminima

Index: 6 Nota: G Figura: Seminima

Index: 7 Nota: A Figura: Seminima

Index: 8 Figura: BarraCompasso

Index: 9 Nota: A Figura: Seminima

Index: 10 Nota: G Figura: Seminima

Index: 11 Nota: F Figura: Seminima

Index: 12 Nota: E Figura: Seminima

Index: 13 Figura: BarraCompasso

Compilando código ABC, este processo pode ser demorado, por favor aguarde

Imagem original

1

3

Exemplo ScoreReader

Transcrito por protótipo Score Reader Alan Soares Carneiro

FURB
UNIVERSIDADE DE BLUMENAU

Fonte: elaborado pelo autor.

Resultado (Música)



Exemplo ScoreReader

Transcrito por protótipo Score Reader Alan Soares Carneiro



Falhas

Caso onde não foi possível detectar nome da nota ou altura do som



Fonte: elaborado pelo autor.

Comparação com correlatos

Características	Correlatos	Mauricenz (2013)	Castro (2014)	Jesus (2015)	Silva (2017)	ScoreReader (2019)
Faz uso da etapa de remoção das linhas da pauta	Sim	Sim	Sim	Sim	Sim	Sim
Reconstrução de partitura a partir de imagem	Não	Não	Não	Sim	Sim	Sim
Utiliza IA (Inteligência artificial) para classificação dos símbolos	Não	Sim	Não	Sim	Sim	Sim
Suportam partituras com instrumento de mais de uma voz	Não	Sim	Não	Sim	Não	
Permite criar modelo de treinamento no protótipo	Não	Não	Não	Não	Sim	Sim
Permitem utilizar conjunto de treino externo	Não	Não	Não	Sim	Sim	Sim
Utiliza linguagem intermediária para geração da partitura	Não	Não	Não	Sim	Sim	Sim
Exportação da partitura em PDF	Não	Não	Não	Não	Sim	Sim
Exportação da partitura em MIDI	Não	Não	Sim	Não	Sim	Sim
Exportação da partitura para MUSICXML	Não	Não	Não	Sim	Sim	Sim

Fonte: elaborada pelo autor.

Dificuldades

- Segmentação de imagens geradas por diferentes editores e com ruídos;
- Cada qualidade e tipo de imagem demanda mais ou menos tratamento;
- Código notação ABC precisa estar escrito sintaticamente e semanticamente correto;
- Mais de um modelo de treino para imagens com resoluções diferentes;
- Compilador de notação ABC para PDF havia sido descontinuado.

Conclusão

- Todos os objetivos propostos foram alcançados;
- Promoveu melhorias e evoluções no protótipo estendido;
- Contribuiu com o Interpress permitindo alcançar mais etapas que compõem um OMR;
- Acessibilidade. Formato MUSICXML poderá ser convertido para Braile e MIDI pode ser ouvido;

Conclusão

- Permite transcrição de partituras de forma automática poupando trabalho ao músico;
- Protótipo ainda possibilita geração de saídas em formato SVG e XHTML;
- Permite criar conjunto de treino dentro do protótipo para utilizar em outras partituras;
- Permite treinamento utilizando imagens manuscritas rotuladas externamente através do MUSCIMA++.

Sugestões

- Focar em mais técnicas para segmentação dos símbolos e tratamento da imagem nas etapas iniciais;
- Permitir a entrada de partituras com ruídos ou danificadas pelo tempo;
- Suporte a partituras com instrumentos de mais de uma voz;
- Adicionar suporte a novos símbolos, como ligaduras, fermatas, pontos de aumento dentre outros;

Sugestões

- Permitir suporte a partituras escritas em notação musical antiga (Universo menor);
- Possibilitar exportar o código compilado para impressão em formato braile mas dentro do protótipo.
- Controle de armazenamento dos arquivos PDF, MIDI e MUSICXML e Rotulações gerados.

Demonstração de funcionamento