FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Extracção de Informação dos Boletins de Saúde Infantil e Juvenil

David Rafael Silva Falcão



Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Rui Camacho

Segundo Supervisor: Luís Teixeira

3 de Agosto de 2020

Extracção de Informação dos Boletins de Saúde Infantil e Juvenil

David Rafael Silva Falcão

Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo Júri:

Presidente: Prof. João Moreira
Arguente: Prof. Carlos Ferreira
Vogal: Prof. Rui Camacho

3 de Agosto de 2020

Resumo

Numa era cada vez mais tecnológica, com facilidade no acesso à Internet, evolução dos assistentes pessoais, entre outros, é tomado como garantido o rápido acesso à informação. Porém nem toda a informação importante se encontra informatizada e os boletins de saúde infantil e juvenil são exemplo disso.

Em Portugal, os registos sobre a evolução dos utentes do 0 aos 20 anos de idade, são documentados em boletins de saúde individuais, acima referidos. Toda a informação presente nestes é manuscrita, tanto sobre a forma de texto, tabelas ou gráficos e, apesar de seguir regras de padronização, está sujeita a falhas humanas. Isto é, o texto nas tabelas pode passar os limites das células, os pontos nos gráficos podem deduzir a mais que um valor, entre outros.

Não há como negar que os dados presentes nos boletins de saúde infantil e juvenil são de extrema importância, quer para a área da pediatria quer para estudos populacionais. Assim sendo, a informatização dos mesmos é uma necessidade. Porém, esta tarefa é trabalhosa e morosa feita manualmente, necessitando de uma elevada quantidade mão de obra.

A solução para a informatização dos boletins passa pela automação de um sistema de extracção da informação. Será feito com o recurso a técnicas de processamento de imagem, reconhecimento ótico de caracteres (*Optical Character Recognition - OCR*), classificação de aprendizagem computacional e mineração de dados. Tal processo envolveria intervenção apenas na digitalização dos boletins e na fase final de classificação dos resultados obtidos, confirmando ou rejeitando os tais, de modo a treinar o sistema e, assim, obter melhores resultados futuramente.

Findada a extracção dos dados, estes serão disponibilizados numa plataforma digital para melhorar drasticamente a facilidade e rapidez de acesso a estes.

Abstract

In an increasingly technological era, with easy access to the Internet, the evolution of personal assistants, among others, quick access to information is taken for granted. However, not all important information is computerized and personal child health records are an example of this.

In Portugal, records on the evolution of children from 0 to 20 years of age are documented in individual health bulletins, referred to above. All the information present in these is handwritten, whether in the form of text, tables or graphs and, despite following rules of standardization, is subject to human error. That is, the text in the tables can pass the cell limits, the points in the graphs can be deducted from more than one value, among others.

There is no denying that the data present in the child and youth health bulletins are extremely important, both for the area of paediatrics and for population studies. Therefore, their computerization is a necessity. However, this task is laborious and time consuming done manually, requiring a high amount of manpower.

The solution for the computerization of newsletters is the automation of an information extraction system. It will be done using image processing techniques, optical character recognition (OCR), machine learning classification and data mining. Such a process would involve intervention only in the digitalization of the bulletins and in the final phase of classification of the results obtained, confirming or rejecting those, to train the system and, thus, obtain better results in the future.

Once the data extraction is finished, it will be made available on a digital platform to drastically improve the ease and speed of access to it.

Agradecimentos

No momento da escrita desta dissertação, vive-se uma pandemia causada pelo vírus SARS-CoV-2. Como tal, um **MUITO OBRIGADO** a todos os profissionais de saúde que estão diariamente na linha da frente contra este coronavírus. São estes os principais responsáveis pela estabilização da pandemia.

Um enorme obrigado também para a minha família por me ter proporcionado todas as condições durante todo o meu percurso académico e em especial neste momento de maior dificuldade para todos.

Gostaria também de agradecer a todos os meus amigos pelo suporte e motivação em todos os momentos, por me fazerem dar sempre mais um pouquinho mais de mim.

Por fim, gostaria de agradecer a todos envolvidos na elaboração desta dissertação, direta ou indirectamente. Ao meu orientador, Rui Camacho, ao coorientador, Luís Teixeira, e a todos os professores da FEUP que sempre se mostraram prontos a ajudar. Um agradecimento também ao Fábio Amarante pelo trabalho elaborado previamente no âmbito desta dissertação.

David Falcão

"When something is important enough,
you do it even if the odds are not in your favor."

Elon Musk

Conteúdo

1 Introdução **1**
- 1.1 Contexto . . . 1
- 1.2 Descrição do problema . . . 2
- 1.3 Motivação . . . 2
- 1.4 Objetivos . . . 2
- 1.5 Estrutura do documento . . . 3

2 Estado da Arte **5**
- 2.1 Trabalhos Relacionados . . . 5
- 2.2 Processamento de Imagem . . . 6
  - 2.2.1 Processamento de Baixo Nível . . . 8
  - 2.2.2 Processamento de Nível Intermédio . . . 9
  - 2.2.3 Processamento de Alto Nível . . . 10
- 2.3 Reconhecimento Ótico de Caracteres . . . 10
- 2.4 Mineração de Dados . . . 14
  - 2.4.1 Objetivos da Mineração de dados . . . 15
  - 2.4.2 Classificação . . . 15
  - 2.4.3 Metodologias de Mineração de Dados . . . 16
- 2.5 Sumário do Capítulo . . . 17

3 Metodologia e Implementação **19**
- 3.1 Metodologia . . . 19
- 3.2 Implementação . . . 21
  - 3.2.1 Divisão por página . . . 21
  - 3.2.2 Detecção do género . . . 24
  - 3.2.3 Isolar conteúdo da página . . . 25
  - 3.2.4 Páginas com gráfico . . . 27
  - 3.2.5 Páginas com tabela . . . 35
  - 3.2.6 Cruzar dados de ambas as fontes . . . 38
  - 3.2.7 Exportação dos dados . . . 39

4 Casos de estudo **41**
- 4.1 Detecção de Pontos . . . 41
  - 4.1.1 Boletins azuis . . . 41
  - 4.1.2 Boletins rosa . . . 43
  - 4.1.3 Conclusões . . . 45
- 4.2 Extracção de informação das tabelas . . . 46
  - 4.2.1 Identificação das células . . . 46

4.2.2 Segmentação e classificação dos caracteres 46

5 Interface de utilizador **49**
5.1 Utilizadores 49
5.2 Funcionalidades 49
5.2.1 Visitantes 49
5.2.2 Utilizadores autenticados 51
5.2.3 Administradores 54
5.2.4 Ver histórico de alterações 54
5.2.5 Eliminar utilizadores 54
5.3 Tecnologias 54

6 Conclusões e Trabalho Futuro **55**
6.1 Trabalho Realizado 55
6.2 Trabalho Futuro 56

Referências **57**

Lista de Figuras

1.1 Boletins de Saúde Infantil e Juvenil 1

2.1 Evolução da literatura sobre detecção de palavras [GSGN17] 5
2.2 Ilustração do conceito de vizinhança [Nun06] 7
2.3 Exemplo de *splitting* [Nun06] 9
2.4 Componentes de um sistema de OCR [CMBK17] 11
2.5 Mineração de dados como uma etapa no processo de descoberta de conhecimento [HKP11] 14
2.6 Objetivos da Mineração de dados [JP05] 15
2.7 Fases da metodologia CRISP-DM [JP05] 17

3.1 Fluxograma da aplicação 19
3.2 Pré-processamento 20
3.3 Processamento 20
3.4 Exemplo de uma página de um PDF 22
3.5 Detecção de bordas de *Canny* 23
3.6 Detecção de bordas e de inversão de página aplicada à figura 3.4 23
3.7 Correcção de rotação 24
3.8 Detecção do género, aplicada à imagem 3.4 25
3.9 Remoção do fundo branco 25
3.10 Realce do conteúdo de uma página 26
3.11 Detecção do conteúdo de uma página 27
3.12 Conteúdo de uma página 27
3.13 Resultado da máscara de cor azul aplicada ao gráfico 3.12a 28
3.14 Detecção dos eixos do gráfico 3.12a 29
3.15 Detecção da escala dos eixos 30
3.16 Detecção dos pontos: gráfico original 30
3.17 Detecção dos pontos por cor: máscaras iniciais 31
3.18 Detecção dos pontos por cor: máscara final 31
3.19 Detecção dos pontos por cor: resultado 32
3.20 Detecção dos pontos por *template matching*: *templates* 32
3.21 Detecção dos pontos por *template matching*: resultado 33
3.22 Detecção dos pontos: boletim azul 33
3.23 Detecção dos pontos por cor: máscaras obtidas 34
3.24 Detecção da grelha de uma tabela 36
3.25 Isolamento de célula da tabela 36
3.26 Detecção de caracteres 37
3.27 Estrutura da base de dados 39

4.1 Boletim azul: deteção de pontos - caso fácil 42
4.2 Boletim azul: deteção de pontos - caso difícil 43
4.3 Boletim rosa: deteção de pontos - caso fácil 44
4.4 Boletim rosa: deteção de pontos - caso difícil 45
4.5 Identificação das células da tabela - casos fácil e difícil 46
4.6 Segmentação de caracteres - caso fácil 47
4.7 Caracteres resultantes de segmentação - caso fácil 47
4.8 Segmentação de caracteres - caso difícil 47
4.9 Caracteres resultantes de segmentação - caso difícil 48

5.1 Mockup: Login 50
5.2 Mockup: Registo 50
5.3 Mockup: Importar boletins 51
5.4 Mockup: Estado boletim 51
5.5 Mockup: Validar dados 52
5.6 Mockup: Analisar dados 53
5.7 Mockup: Treinar classificadores 53

Lista de Tabelas

2.1 Estudos recentes [Ama18] 6

4.1 Boletim azul: deteção de pontos - caso fácil (resultados) 42
4.2 Boletim azul: deteção de pontos - caso fácil (desvios) 42
4.3 Boletim azul: deteção de pontos - caso difícil (resultados) 43
4.4 Boletim rosa: deteção de pontos - caso fácil (resultados) 44
4.5 Boletim rosa: deteção de pontos - caso fácil (desvios) 44
4.6 Boletim rosa: deteção de pontos - caso fácil (resultados) 45

Acrónimos

BSIJ	Boletins de Saúde Infantil e Juvenil
CNN	Convolutional Neural Network
HMM	Hidden Markov Modeling
MNIST	Modified National Institute of Standards and Technology
OCR	Optical Character Recognition
PDF	Portable Document Format
PNG	Portable Network Graphics
RNA	Rede Neural Artificial

Capítulo 1

Introdução

Neste capítulo é feita uma introdução aos Boletins de Saúde Infantil e Juvenil, bem como uma breve explicação do seu contexto na área da saúde em Portugal. Posteriormente, são apresentados a motivação que levou ao desenvolvimento desta dissertação e os objectivos com a sua realização. Para complementar, é esclarecida a estrutura deste documento.

1.1 Contexto

O Boletim de Saúde Infantil e Juvenil, abreviado para BSIJ, destina-se ao registo dos dados mais importantes relacionados com a saúde da criança ou jovem ([FSF11]), como, por exemplo, estatura, peso, perímetro cefálico, índice de massa corporal, entre outras informações, bem como algumas relações entre estes.

> "O Boletim de Saúde Infantil e Juvenil destina-se ao registo dos factos mais importantes relacionados com a saúde da criança e do jovem." [dS]



Figura 1.1: Boletins de Saúde Infantil e Juvenil

Existem diversas versões dos BSIJ, duas delas apresentadas na figura 1.1 (à esquerda para rapaz, à direita para rapariga), porém as alterações não são muito significativas. O que é invariável ao longo de todas as versões, é o facto da informação contida nos boletins se encontrar manuscrita, tanto sobre a forma de tabelas, gráficos ou apontamentos.

1.2 Descrição do problema

Nos anos que decorrem, com o constante desenvolvimento tecnológico, há uma enorme facilidade e rapidez em acesso à informação. Assim sendo, o facto dos dados dos BSIJ estarem manuscritos em livros individuais é uma excepção a este desenvolvimento.

Assumindo a necessidade de extrair estes dados para o formato digital, a primeira abordagem seria a conversão manual dos boletins. Porém, devido ao enorme número de BSIJ em circulação e à grande quantidade de informação presente em cada boletim, este processo iria ser moroso e exigir muita mão de obra e, consequentemente, conduzir a custos elevados.

Por outro lado, temos a automação de um sistema para extrair a informação dos boletins. O maior desafio desta abordagem é a complexidade do processo, tendo em conta que a informação pode estar presente em texto corrido, gráficos ou tabelas. Outro grande desafio é o reconhecimento da caligrafia.

1.3 Motivação

A motivação por detrás desta dissertação está nos desenvolvimentos a nível da pediatria que esta poderá trazer.

Os trabalhos de investigação na área da pediatria têm por base uma enorme quantidade de dados e os Boletins de Saúde Infantil e Juvenil são uma fonte preciosa destes. A extracção destes dados levaria a uma maior facilidade de acesso a estes e, consequentemente, impulsionaria os trabalhos de investigação na área.

1.4 Objetivos

Esta dissertação tem objetivos muito concretos, que podem ser resumidos como:

- automatizar o processo de extracção da informação dos Boletins de Saúde Infantil e Juvenil;
- desenvolver uma plataforma para a visualização dos dados obtidos.

Assim sendo, o primeiro objetivo passa pela elaboração de um sistema quase autónomo (devido à sensibilidade dos dados é preciso uma validação à posteriori) que faça a extracção dos dados dos BSIJ de forma rápida e que combata a elevada necessidade de mão de obra da conversão manual dos boletins.

Concluída a extracção, o objetivo seguinte passa elaboração de uma plataforma para a visualização dos dados obtidos e, assim, combata a dificuldade de acesso a estes, situação descrita na secção 1.2.

1.5 Estrutura do documento

Para além da introdução, este documento apresenta mais cinco capítulos, sendo estes:

- **Estado da Arte** que contem os resultados da pesquisa sobre implementações com objetivos semelhantes e sobre as suas metodologias;
- **Metodologia e Implementação** onde é apresentada a metologia utilizada e a sua implementação na resolução desta dissertação;
- **Casos de Estudo** onde são apresentadas, analisadas e comparados os resultados de execuções em diferentes cenários;
- **Interface de utilizador** onde é apresentada a plataforma de acesso aos dados extraídos;
- **Conclusões e Trabalho Futuro** que apresenta, de breve, os resultados do trabalho desenvolvido durante esta dissertação e aspectos que poderiam ser melhorados futuramente.

Capítulo 2

Estado da Arte

Neste capítulo é feita uma apresentação do estado da arte e são apresentados trabalhos relacionados com esta dissertação.

De notar que esta dissertação é a continuação do desenvolvimento da dissertação de mestrado do aluno Fábio Amarante [Ama18]. Deste modo, tem por base o trabalho e pesquisa já desenvolvido.

2.1 Trabalhos Relacionados

A extracção de informações presentes em documentos físicos, tais como, por exemplo, documentos históricos, cheques bancários, relatórios e receitas médicas, é uma área que tem atraído um interesse crescente ao longo dos anos. E, com este interesse, têm surgido novas técnicas para fazer a extracção, consoante o tipo de informação, de linguagem e do documento [GSGN17]. É possível ver essa evolução na imagem 2.1.

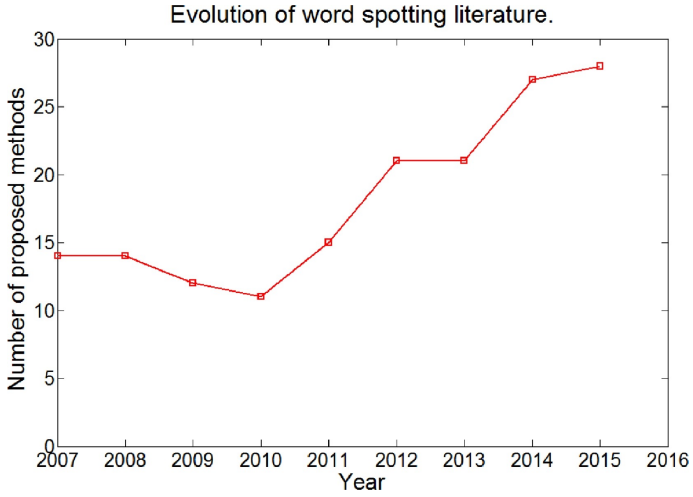


Figura 2.1: Evolução da literatura sobre detecção de palavras [GSGN17]

Na tabela 2.1, são apresentados alguns estudos relacionados com a extracção de caracteres de documentos manuscritos. A tabela foi retirada de [Ama18], porém ainda se encontra de acordo com a realidade atual, ou seja, apresenta soluções que ainda são consideradas eficazes na actualidade.

Tabela 2.1: Estudos recentes [Ama18]

Artigo	**Ano**	**Métodos de Extracção**	**Métodos de Representação**	**Área de Aplicação**
[RBD+17]	2017	*Pyramid Histogram of Oriented Gradient (PHOG) + Tandem*	Hidden Markov Modeling (HMM)	Saúde
[RLSG16]	2016	*Projections of Oriented Gradients Variants*	-	Documentos Manuscritos
[SF16]	2016	*Pyramid Histogram of Oriented Characters (PHOC)*	*Convolutional Neural Network (CNN)*	Documentos Manuscritos
[AG14]	2014	*Discrete Cosine Transform (DCT)*	HMM	Reconhecimento de Dígitos Manuscritos

De acordo com os artigos selecionados, entre outros, as opções a serem mais aprofundadas serão as abordagens com métodos HMM e redes neuronais.

2.2 Processamento de Imagem

Processamento de imagem consiste em transformar uma imagem original de modo a que se consiga obter uma baseada nesta, mas realçando um conjunto de características de interesse.

> "Ao vasto conjunto de operações que podemos aplicar em uma matriz que representa uma imagem denominamos processamento de imagem."[Nun06]

A matriz que representa a imagem é constituída pelo conjunto de pixeis que, quando agrupados, formam essa mesma imagem.

O pixel é a menor unidade sobre a qual podemos realizar transformações. Entre si, os pixeis apresentam alguma relações, necessárias para a realização de transformações sobre a matriz-imagem. Essas relações são:

- **Vizinhança**
  Segundo [GW07], a vizinhança de um pixel pode ser definida da seguinte forma: seja p, um pixel nas coordenadas (x,y):

- A vizinhança de 4 do pixel *p* (*N4(p)*) é composta por seus vizinhos de coordenadas: *(x+1,y)*, *(x-1,y)*, *(x,y+1)* , *(x,y-1)*.
- A vizinhança diagonal do pixel *p* (*ND(p)*) é composta por seus vizinhos de coordenadas: *(x+1,y+1)*, *(x+1,y-1)*, *(x-1,y+1)*, *(x-1,y-1)*.
- A vizinhança diagonal de 8 do pixel *p* (*N8(p)*) é composta pelo conjunto de todos os pixeis vizinhos, ou seja, *N4(p)* ∪ N8(p)

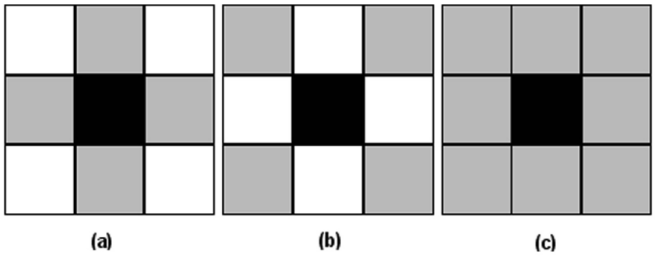


Figura 2.2: Ilustração do conceito de vizinhança [Nun06]

Atendendo à figura 2.2 e tomando o ponto central como pixel de interesse e os pixeis da vizinhança os pixeis coloridos, o caso (a) representa *N4(p)*, (b) representa *ND(p)* e (c) representa *N8(p)*

- **Adjacência**
  A adjacência é característica de um par de pixeis vizinhos que compartilham uma borda ou um vértice, sendo que:
  - um par de pixeis que compartilha uma borda é dito "adjacente por borda"ou "4-adjacente";
  - um par de pixeis que compartilha um vértice é dito “adjacente por vértice” ou “8-adjacente”.

    Na figura 2.2, tomando o caso (a) como exemplo, observa-se que os pixeis cinzentos são 4-adjacentes e também 8-adjacentes do pixel de preto, pois partilham uma borda e um vértice com este pixel. Na mesma figura, os pixeis brancos são somente 8-adjacentes do pixel preto, pois compartilham somente um vértice com o mesmo.

- **Conectividade**
  A conectividade é um conceito utilizado para estabelecer bordas de componentes numa imagem. Dois pixeis são conectados se cumprirem ambos os seguintes requerimentos:
  - são adjacentes;
  - obedecem a um critério de similaridade dentro de uma escala de cor.

- **Distância entre pixeis**
  A distância entre pixeis é um valor mensurável, constituindo uma importante definição para

grande parte dos algoritmos que manipulam a imagem. A formula para calcular essa distância vai ser adaptada consoante a aplicação especifica.

Aplicando os conceitos anteriores a processamento de imagens, este pode dividir-se em três níveis, consoante a sua função específica:

- **processamento de baixo nível:** operações primitivas no nível de pixel, como redução de ruído, correções geométricas, equalização de histograma, etc;
- **processamento de nível intermédio:** análise de imagens, como tarefas de segmentação e/ou classificação de objetos individuais;
- **processamento de alto nível:** estuda a relação de objetos com o ambiente e/ou com outros objetos apresentados na imagem.

2.2.1 Processamento de Baixo Nível

Atendendo ao princípio que todos os pixeis de uma vizinhança numa imagem têm aproximadamente/exatamente os mesmo parâmetros físicos, através de processamento de baixo nível é possível explorar a redundância nessa vizinhança. Assim é possível, por exemplo, retirar amostras de ruído, resolver descontinuidades de uma superfície, entre outros. [Nun06] [Bha14]

Algumas técnicas de processamento aplicadas neste sentido são as seguintes:

- **Alterações globais no brilho:** é possível transformar uma imagem mais brilhante ou menos brilhante (mais clara ou mais escura) através da soma ou subtração, respetivamente, de uma constante em todos os pixeis da imagem. A variância no brilho é tanto maior quanto o valor absoluto dessa constante.
- **Quantização do histograma:** é a redução da quantidade de níveis distintos de cinza numa imagem. Um exemplo da sua utilização é a remoção de gradações indesejáveis numa imagem, promovendo um efeito de aumento de contraste.
- ***Splitting*:** visa aumentar o contraste de uma imagem com base no seu histograma. Com este fim, os pixeis são divididos em dois grupos distintos de níveis de cinza, de modo que os pixeis com valores menores são diminuídos e os com valores mais altos são aumentados.

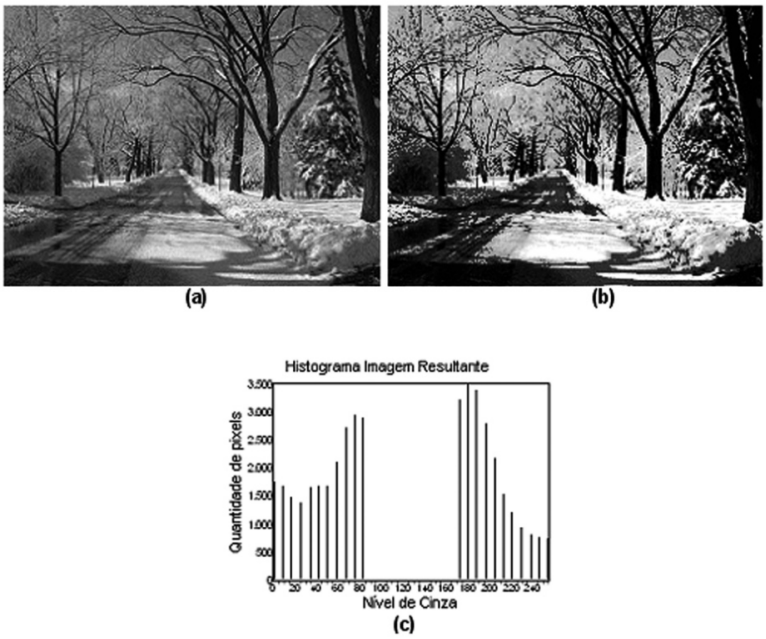


Figura 2.3: Exemplo de *splitting* [Nun06]

A imagem 2.3 apresenta um exemplo de *splitting* em que: (a) imagem original, (b) imagem resultante e (c) histograma resultante.

- **Equalização:** tem a finalidade de obter um histograma uniforme, permitindo uma redução das diferenças acentuadas e, assim, acentuando detalhes não visíveis anteriormente. É uma técnica comum no pré-processamento em aplicações de reconhecimento de padrões.

- **Média da vizinhança:** consiste na geração de uma imagem na qual o nível de cinza de cada pixel é obtido através da média dos valores de cinza dos pixeis de uma vizinhança pré-definida. É extremamente eficaz para a eliminação de ruído porém, ao ser aplicada, provoca uma perda de nitidez na imagem e, assim, perda de definição de bordas.

2.2.2 Processamento de Nível Intermédio

Esta fase de processamento tem o objetivo de identificar as formas significativas e isolá-las, com a finalidade de fornecer informações sobre a sua interpretação.

Fazendo uma análise sobre o ponto de vista dos objetivos desta dissertação, processamento deste nível irá ser aplicado para segmentar texto e gráficos. Deste modo, irão ser esses pontos a focar de seguida.

2.2.2.1 Segmentação de gráficos

Analisando [SMS12], os métodos de segmentação de imagens podem ser divididos em:

- **Métodos baseados na região:** levam em consideração a continuidade presente na imagem, ou seja, eles dividem a imagem em sub-regiões, dependendo de algumas regras. Por exemplo, levando em consideração o nível de cinza, é possível delimitar uma região cujos pixeis têm o mesmo valor. O objetivo principal é definir regiões na imagem de acordo com suas funções anatómicas ou funcionais.
- **Métodos baseados em bordas:** baseiam-se em descontinuidades, encontrando mudanças bruscas no valor de intensidade. São úteis para a segmentação de linhas, por exemplo.

2.2.2.2 Segmentação de texto

Com base em [MAN+14], existem três níveis para alcançar a segmentação da imagem de texto. Esse níveis são:

1. **Segmentação de linhas:** segmentação do texto linha por linha em que o espaço vertical em branco entre duas linhas é usado para fazer essa divisão;
2. **Segmentação de palavras:** segmentação de palavras de uma linha, através da análise do espaçamento horizontal entre palavras numa linha;
3. **Segmentação de caracteres:** por fim, segmentação de caracteres individuais de uma palavra. É o nível mais desafiador tendo em conta a possibilidade de sobreposição de alguns caracteres quando manuscritos.

2.2.3 Processamento de Alto Nível

O último nível de processamento de uma imagem tem o objetivo de perceber a imagem, o assunto desta. Isto é feito através do reconhecimento dos diferentes conteúdos segmentados e extraídos através de métodos de processamento de nível intermédio, abordados em 2.2.2.

Um dos focos desta dissertação é o reconhecimento de caracteres, na qual esta fase desempenha um papel importante no reconhecimento de padrões numa imagem. Será portanto retratada em mais detalhe na secção 2.3, na componente **Treino e reconhecimento**.

2.3 Reconhecimento Ótico de Caracteres

> "Optical Character Recognition (OCR) has become an important and widely used technology. Among its many practical applications are the scanners used at store check-out counters, money changing machines, office scanning machines, and the efforts to automate the postal system."[MNY99]

Reconhecimento Ótico de Caracteres(OCR, do inglês, *Optical Character Recognition*) é uma tecnologia para reconhecer caracteres a partir de um arquivo de imagem e que se tem vindo a desenvolver ao longo dos anos, sobretudo devido ao aumento do poder de processamento dos dispositivos e à aplicabilidade em diversas áreas.

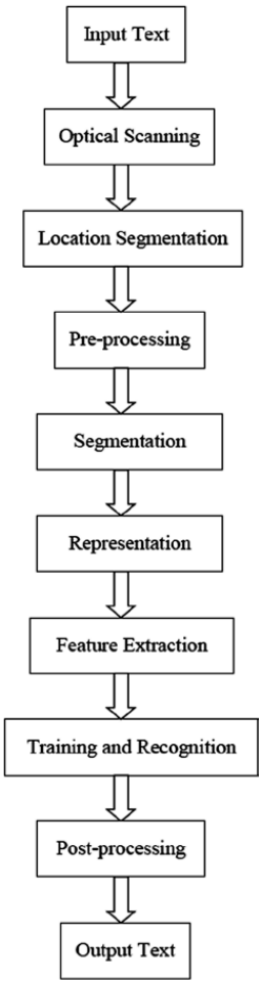


Figura 2.4: Componentes de um sistema de OCR [CMBK17]

A figura 2.4 apresenta os diferentes componentes de um sistema de OCR, desde o texto de entrada, em formato de imagem, até ao texto de saída. Os mesmos podem ser sintetizados da seguinte forma, de acordo com [CMBK17]:

- **Digitalização óptica:** processo de transformação de uma imagem(código analógico) em código digital.
- **Segmentação localizada:** a segmentação determina os constituintes de uma imagem. É necessário localizar regiões do documento que tenham dados impressos e que são distinguíveis de figuras e gráficos.
- **Pré-processamento:** os dados estão sujeitos a várias etapas preliminares de processamento para torná-los utilizáveis nas etapas seguintes. Os principais objetivos do pré-processamento são:
  - **redução de ruído:** o ruído introduzido pelo dispositivo óptico de digitalização ou pelo o instrumento de escrita causa segmentos de linha desconectados, solavancos e lacunas

nas linhas, etc. É necessário eliminar essas imperfeições antes do processamento real dos dados.

  - **normalização dos dados:** os métodos de normalização visam remover as variações da escrita e obter dados padronizados.
  - **compressão:** a compactação para OCR requer técnicas de domínio de espaço para preservar as informações de forma. As duas técnicas de compressão populares usadas são: limiar e afinamento.

- **Segmentação:** separação de um caractere nos seus componentes. Esta etapa afeta positivamente a taxa de reconhecimento.

- **Representação:** no caso mais simples, o nível de cinza ou imagens binárias passam a um reconhecedor. No entanto, na maioria dos sistemas de reconhecimento, a fim de evitar complexidade extra e aumentar a precisão dos algoritmos, é necessária uma representação mais compacta e característica. Os métodos de representação de imagem de caracteres são geralmente classificados em três grupos principais:

  - **Transformação global e expansão em série:** um sinal contínuo, geralmente, contém mais informações do que precisa ser representado para fins de classificação. Uma maneira de representar um sinal é por uma combinação linear de uma série de funções simples e bem definidas. Os coeficientes da combinação linear fornecem uma codificação compacta conhecida como transformação ou expansão em série. Deformações como translação e rotações são invariantes sob transformação global e expansão em série.
  - **Representação estatística:** a representação de um caractere pela distribuição estatística de pontos trata de variações de estilo até certo ponto. Embora esse tipo de representação não permita a reconstrução da imagem original, é usado para compressão do conjunto de dados, proporcionando alta velocidade e baixa complexidade.
  - **Representação geométrica e topológica:** as várias propriedades globais e locais dos caracteres podem ser representadas por recursos geométricos e topológicos com alta tolerância a distorções e variações de estilo. Esse tipo de representação também pode codificar algum conhecimento sobre a estrutura do objeto ou fornecer algum conhecimento sobre que tipo de componentes compõem esse objeto.

- **Extracção de recursos:** o objetivo da extracção de recursos é capturar as características essenciais dos símbolos.

- **Treino e reconhecimento:** uso de metodologias de reconhecimento de padrões para fazer a associação de uma amostra desconhecida com uma classe predefinida. As abordagens para reconhecimento de padrões, de acordo com [AYV01] são as seguintes:

  - **Correspondência de modelos:** correspondência dos modelos armazenados com o caractere a ser reconhecido, determinando o grau de semelhança entre dois vetores, como grupo de pixeis, formas, curvatura, etc. As técnicas de correspondência podem ser classificadas em três classes:
    * Correspondência direta;
    * Modelos deformáveis e correspondência elástica;
    * *Relaxation matching*.
  - **Técnicas estatísticas:** A teoria da decisão estatística preocupa-se com a estatística funções de decisão e um conjunto de critérios de optimização que maximizam a probabilidade do padrão observado, dado o modelo de uma determinada classe. As principais abordagens estatísticas aplicadas em reconhecimento de caracteres são:
    * Reconhecimento não paramétrico;
    * Reconhecimento paramétrico;
    * Análise de agrupamento;
    * *Hidden Markov Chains*(HMM);
    * Raciocínio definido difuso.
  - **Técnicas estruturais:** descrição recursiva de um padrão complexo em termos de padrões mais simples, com base na forma do objeto. Supõe-se que as primitivas de caracteres extraídas da escrita sejam quantificáveis e é possível encontrar as relações entre elas. Os métodos aplicados nestes casos são:
    * Métodos gramaticais;
    * Métodos gráficos;
  - **Redes Neurais Artificiais (RNAs):** as RNAs são sistemas adaptativos de processamento de informações não-lineares que têm como principais características auto-adaptação, auto-organização e aprendizado em tempo real. As RNAs possui uma arquitectura massivamente paralela, de modo que executa computação numa taxa mais alta em comparação às técnicas clássicas. Elas adaptam-se às alterações nos dados e aprendem as características do sinal de entrada. As RNAs contém muitos nós e a decisão final depende da interação complexa de todos os nós. Apesar dos diferentes princípios subjacentes, pode ser demonstrado que a maioria das arquitecturas da RNA é equivalente a métodos estatísticos de reconhecimento de padrões.

- **Pós-processamento:** algumas das atividades de pós-processamento mais usadas incluem agrupamento e detecção e correção de erros.

Concluídos todos estes processos, obtemos, a partir de texto embutido numa imagem, texto editável.

2.4 Mineração de Dados

A mineração de dados é formada por um conjunto de ferramentas e técnicas que, através do uso de algoritmos de aprendizagem ou classificação baseados em redes neurais e estatística, são capazes de explorar um conjunto de dados, extraindo ou ajudando a evidenciar padrões nestes dados e auxiliando na descoberta de conhecimento.

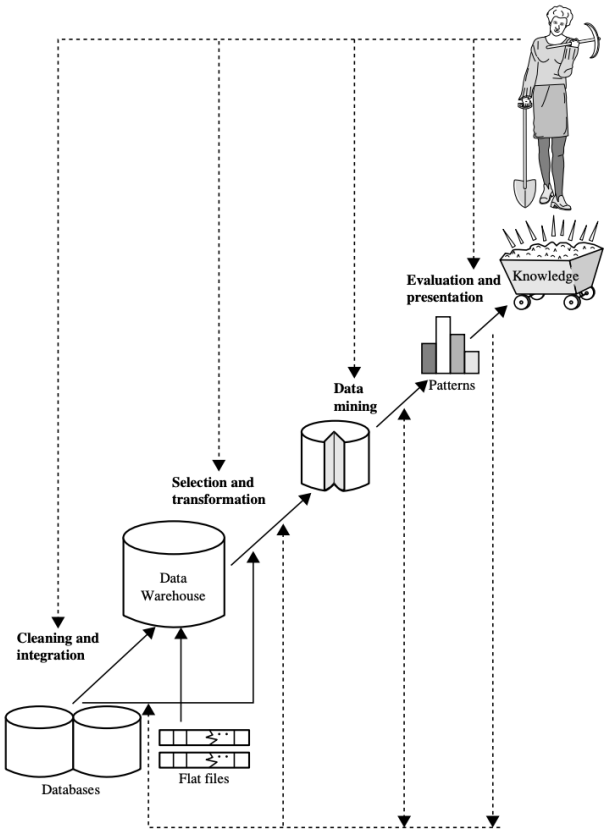


Figura 2.5: Mineração de dados como uma etapa no processo de descoberta de conhecimento [HKP11]

Segundo [HKP11] e tendo por análise a figura 2.5, os passos para a descoberta de conhecimento são os seguintes:

1. **Limpeza dos Dados:** remoção de ruído e de dados inconsistentes;
2. **Integração de Dados:** combinação de várias fontes de dados;
3. **Seleção de Dados:** seleção dos dados importantes para a tarefa de análise;
4. **Transformação de Dados:** transformação e consolidação dos dados de forma apropriada para a mineração, executando operações de resumo ou agregação;
5. **Mineração de Dados:** extracção de padrões de dados através da aplicação de métodos inteligentes;

6. **Avaliação de padrões:** identificação dos padrões realmente interessantes que representam o conhecimento, com base em medidas de interesse;

7. **Apresentação de conhecimento:** utilização de técnicas de visualização e representação do conhecimento para apresentar o conhecimento extraído.

2.4.1 Objetivos da Mineração de dados

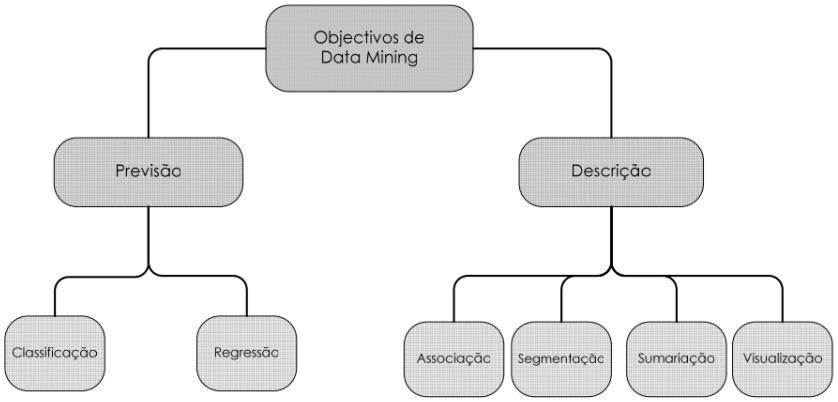


Figura 2.6: Objetivos da Mineração de dados [JP05]

Os objectivos de Mineração de Dados podem ser combinados em duas categorias, nomeadamente em Previsão e Descrição. A Classificação e Regressão enquadram-se na primeira, enquanto que a Associação, Segmentação, Sumariação e Visualização se enquadram na segunda, tal como apresentado na figura 2.6.

Nesta dissertação irá ser a abordada apenas a previsão, e especificamente a componente Classificação, tendo em conta a necessidade de implementação de algoritmos de classificação para reconhecimento de caracteres manuscritos.

2.4.2 Classificação

A tarefa de classificação consiste em construir um modelo de algum tipo que possa ser aplicado a dados não classificados visando categorizá-los em classes. Um objecto é examinado e classificado de acordo com uma classe definida.

Alguns dos métodos utilizados na tarefa de classificação são os seguintes:

- Redes neuronais artificiais
- Classificadores de Bayes
- Máquina de vetores de suporte
- Modelos ocultos De Markov

- K-ésimo vizinho mais próximo
- Aprendizagem profunda

2.4.2.1 Redes neuronais artificiais

As redes neuronais artificiais são sistemas adaptativos de processamento de informações não-lineares que têm como principais características auto-adaptação, auto-organização e aprendizagem em tempo real. Receberam o nome de redes neuronais, graças à semelhança com o cérebro humano. Elas são usados para várias tarefas, como classificação, regressão, modelagem e previsão e estão em grande expansão devido ao aumento constante do uso dessas técnicas por diferentes áreas de aplicação.

2.4.2.2 HMM

HMM é uma máquina de estados probabilísticos que segue um processo de Markov com parâmetros desconhecidos e com um conjunto de estados discretos e finitos, em que o objetivo é determinar os parâmetros ocultos dos parâmetros observáveis. HMM satisfaz a propriedade da cadeia de Markov na qual o próximo estado depende do estado atual.

2.4.3 Metodologias de Mineração de Dados

O processo de minerar dados é uma tarefa muito complexa. Assim sendo, é imprescindível o uso de uma metodologia completa e sistemática. Dois exemplos de metodologias para este processo são a metologia a metodologia CRISP-DM (*CRossIndustry Standard Process for Data Mining*) e a metodologia SEMMA (*Sample, Explore, Modify, Model,Assessement*).

Olhando para ambas as metodologias, a metodologia CRISP-DM encontra-se mais completa, incorporando as fases de Estudo do Negócio, Estudo dos Dados e Implementação, que a metodologia SEMMA não incorpora. Deste modo, nesta dissertação irá ser aprofundada somente a metodologia CRISP-DM.

2.4.3.1 Metodologia CRISP-DM

O CRISP-DM é um modelo de mineração de dados não proprietário e disponível gratuitamente, desenvolvido pelos líderes da indústria. Esse modelo foi desenvolvido para incentivar as melhores práticas e provem às organizações a estrutura necessária para obter resultados melhores e mais rápidos da mineração de dados.

Esta metodologia apresenta seis fases, apresentadas abaixo:

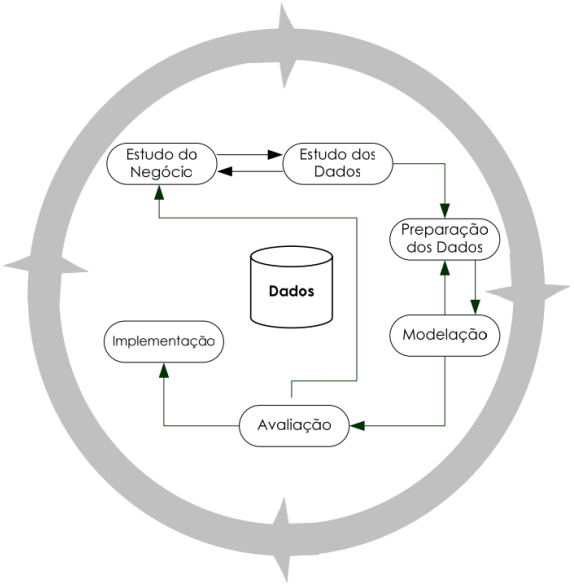


Figura 2.7: Fases da metodologia CRISP-DM [JP05]

1. **Entender o Negócio:** foca em entender o objetivo do projeto a partir de uma perspectiva de negócios, definindo um plano preliminar para atingir os objetivos;

2. **Entender os Dados:** recolha de dados e início de atividades para familiarização com os dados, identificando problemas ou conjuntos interessantes;

3. **Preparação dos Dados:** seleção e limpeza dos dados e integração e formatação dos dados para construir um conjunto de dados;

4. **Modelagem:** aplicação de várias técnicas de modelagem e calibração dos seus parâmetros, tendo em vista a otimização;

5. **Avaliação:** avaliação dos resultados, revisão do processo e determinação dos passos seguintes;

6. **Desenvolvimento:** organização do conhecimento adquirido pelo modelo.

2.5 Sumário do Capítulo

Este trabalho irá englobar várias áreas, porém, todas elas interligadas e dependentes umas das outras.

De técnicas de processamento de imagem, a técnicas de mineração de dados, descritas e analisadas no decorrer deste capítulo, todas as áreas terão contributo para o objetivo final.

Feita a análise do estado da arte e dos trabalhos relacionados, no capítulo 3 irá ser apresentada a metodologia a ser utilizada nesta dissertação.

Capítulo 3

Metodologia e Implementação

Durante este capítulo será descrita a metodologia utilizada nesta dissertação, bem como as principais etapas da sua implementação.

3.1 Metodologia

Visando o objetivo desta dissertação, que é extracção de dados manuscritos de um conjunto de Boletins de Saúde Infantil e Juvenil, foi planeado o fluxograma apresentado na figura 3.1.

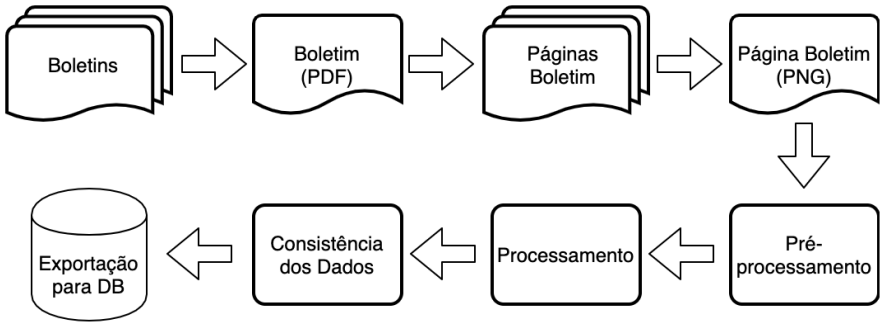


Figura 3.1: Fluxograma da aplicação

Sinteticamente, será abordado cada boletim de forma individual. Em primeiro lugar na *pipeline*, o boletim a ser processado, em formato PDF, é dividido por páginas e salvo num directório com o mesmo nome. Durante o processo de divisão, é também detectado o género da criança detentora do boletins, através das cores azul e rosa presentes nos boletins. O género é importante em etapas seguintes.

De seguida e sequencialmente, cada página passa por um processo de pré-processamento, onde são aplicadas técnicas para remoção de ruído e para enaltecer a informação importante para ser processada. Este pré-processamento varia consoante o tipo de conteúdo da página - gráfico ou tabela. Esta distinção é feita através do *input* do número da página a ser tratada. Este número é

lido na parte inferior da página ou, em caso de falha, tratado como sequencial na pipeline. Na figura 3.2, são apresentadas as etapas mais importantes nesta fase.

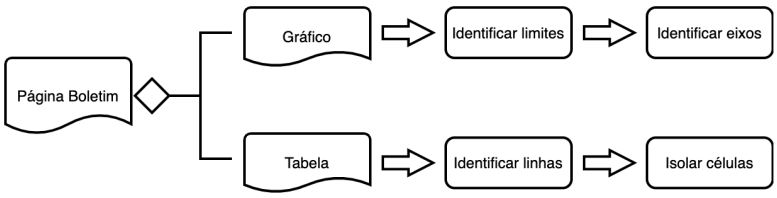


Figura 3.2: Pré-processamento

No caso da página em questão se tratar de um gráfico, e como pode ser analisado na figura 3.2, os passos necessários no pré-processamento são a identificação dos limites do conteúdo e a identificação dos eixos do gráfico.

Caso contrário, se a página se tratar de uma tabela, no pré-processamento é necessário identificar as linhas da tabela e, em seguida, isolar as células.

Posteriormente, dá-se o processamento. Do mesmo modo que o processo anterior, este varia consoante o conteúdo da página a ser tratada. Os passos mais importantes do processamento são apresentados na figura 3.2, a seguir.

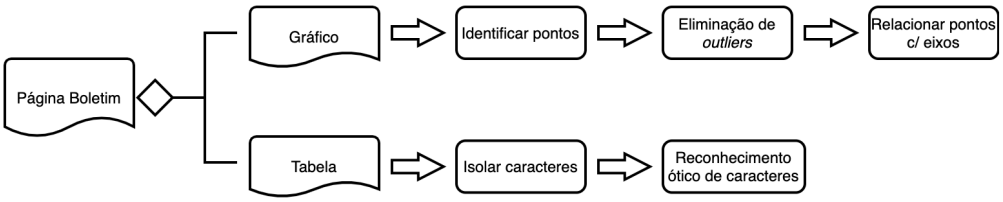


Figura 3.3: Processamento

Tratando-se de um gráfico, e já tendo por base os limites deste e os eixos identificado pela etapa anterior, o objetivo é identificar os pontos e atribuir-lhe as coordenadas reais. É também necessário aplicar um varrimento nos pontos obtidos para eliminar eventuais *outliers*, de modo a garantir que esses valores não tenham influência negativa nos dados reais.

Tratando-se de uma tabela, de célula em célula, é feito o isolamento de caracteres para depois fazer o seu reconhecimento. Ao isolar os caracteres, não só optimiza o seu reconhecimento mas também permite guardar o número de caracteres para, se em caso de falha no reconhecimento, ser possível determinar o intervalo em que se encontra o valor da célula.

Tendo obtido os valores, tanto dos gráficos como das tabelas, e para garantir a consistência dos dados, é feito um cruzamento dos valores de ambas as fontes. Com isto, é possível obter o grau de certeza aproximado de cada valor e, como tal, descartar aqueles que tenham elevado nível de incerteza, considerando-os como outliers.

Como último passo na *pipeline*, e tendo já um conjunto consistente de dados, é feita a exportação dos mesmo para uma base de dados, que garantirá um acesso fácil e rápido aos mesmos. Permitirá também que, olhando para os boletins originais, sejam confirmados os valores obtidos.

Na listagem 1, é possível analisar o pseudo-código que corresponde à pipeline descrita.

```
for boletim in boletins:
    paginas = separar_por_pagina(boletim)

    dados_tabela = None
    dados_graficos = {}

    for pagina in paginas:
        if pagina == "Grafico":
            imagem = pre_processar_grafico(pagina)
            dados = processar_grafico(imagem)
            dados_graficos[nrpagina] = dados

        elif pagina == "Tabela":
            imagem = pre_processar_tabela(pagina)
            dados = processar_tabela(imagem)
            dados_tabela = dados

    dados_consistentes =  cruzar_fontes_dados(dados_tabela, dados_graficos)
    exportar_base_de_dados(dados_consistentes)
```

Listing 1: Pseudo-código

3.2 Implementação

Os tópicos seguintes representam as principais etapas da *pipeline* desenvolvida na realizada nesta dissertação.

O código foi escrito em Python com recurso à biblioteca OpenCv. Esta biblioteca foi escolhida, pela sua abundância em recursos de processamento de imagem, pela sua facilidade de uso e também pela sua excelente documentação.

Para o reconhecimento de caracteres foram utilizadas as bibliotecas Pytesseract (um *wrapper* do *Google's Tesseract-OCR Engine*) e Keras.

3.2.1 Divisão por página

Para começar é necessário obter páginas individuais dos BSIJ. Para tal é necessário proceder a duas divisões: em primeiro, dividir o PDF do boletim em páginas e, em segundo, dividir cada página do PDF em páginas do boletim - cada página do PDF tem, por norma, 2 páginas do boletim, como se vê na figura 3.4.

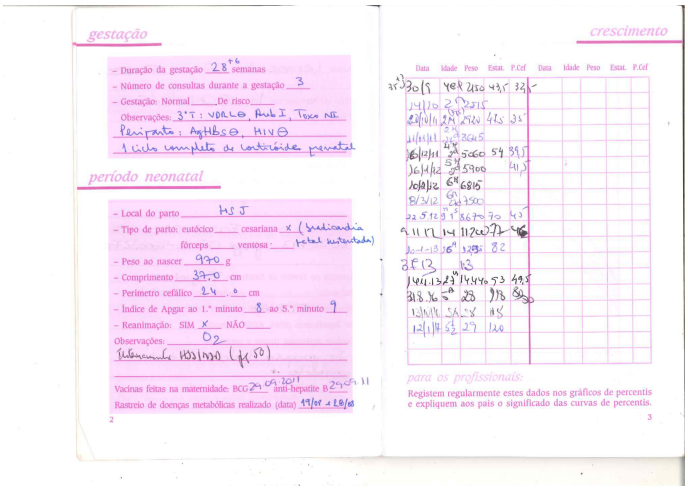


Figura 3.4: Exemplo de uma página de um PDF

A primeira divisão foi direta com o uso do módulo pdf2image, disponível em Belval/pdf2image. Deste modo, o foco será na segunda divisão.

Para proceder à divisão das diferentes páginas do documento presentes na mesma imagem foi tomado como garantido que, na existência de mais que uma página por imagem:

- ambas as páginas têm as mesmas dimensões;
- as páginas estão unidas
- a imagem está em rotação de paisagem, visto não haver espaço necessário para estarem 2 páginas lado a lado em rotação de retrato

Tomando estas afirmações como base, o primeiro passo passa por rodar a imagem de modo que esta fique em modo paisagem. De seguida, a imagem é convertida em escalas de cinzento, é aplicada um filtro gaussiano para reduzir imperfeições e, por fim, o tamanho da imagem é reduzido de modo a optimizar a detecção de bordas de *Canny*. O processo de detecção de bordas de Canny baseia-se em dois critérios básicos de desempenho, i.e., os critérios de detecção e localização. Estes critérios estão sujeitos ainda a um terceiro, conhecido como injunção de resposta múltipla, que força o processo a detectar uma única borda onde existe somente uma borda verdadeira. Para este algoritmo foi utilizada a implementação do OpenCv, ***Canny***. Um exemplo da máscara resultante pode ser observada na imagem 3.5.

Para obter os contornos a partir da máscara obtida, foi implementado um algoritmo que percorre a imagem, das bordas para o centro, e detecta oscilações abruptas na imagem. Estas oscilação representam as bordas dos documentos. Se a área contida por estas bordas for superior a 60% do total da página, a imagem contêm 2 páginas de documento, caso contrário, apenas 1. No caso se serem 2 imagens, a área detectada é dividida a meio na aresta de maior comprimento.

Com objetivo de eliminar outras variáveis que podem induzir a maus resultados, foram tomados passos adicionais. Para continuar é necessário detectar o género à qual pertence o boletim. Para tal, é usada a solução apresentada na secção 3.2.2. Os erros a tentar evitar são:

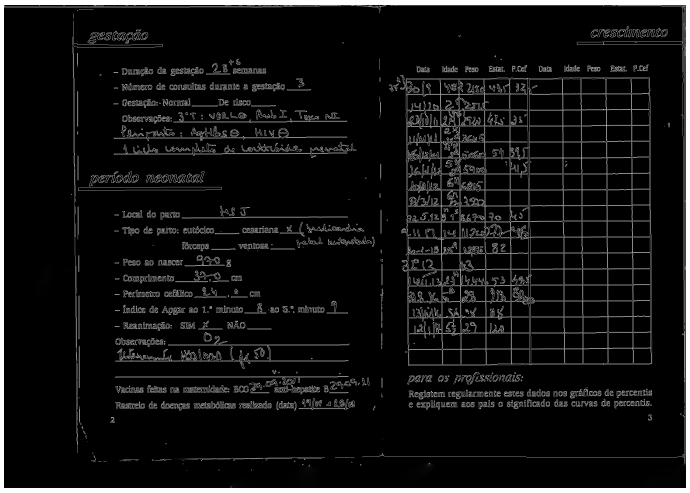
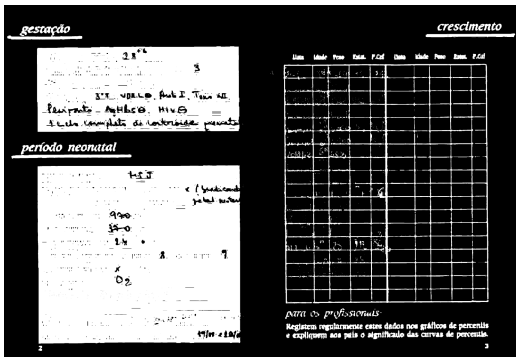


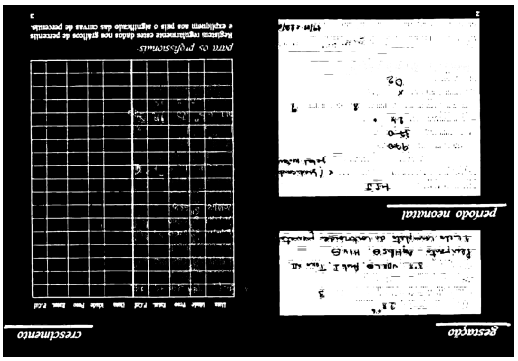
Figura 3.5: Detecção de bordas de *Canny*

- inversão das páginas;
- ligeira rotação das páginas

De modo a controlar o primeiro erro, a solução implementada passa por aplicar a máscara da cor atribuída ao género (já detectado anteriormente) às duas primeiras páginas do documento. Tendo por base que a primeira página é a que apresenta maior intensidade de cor, se o resultado da máscara assim não o demonstrar, significa que as páginas estão rodadas 180º. É apresentado uma execução deste algoritmo na imagem 3.6, em que 3.6a representa uma orientação correta e 3.6b representa uma página invertida. Para corrigir as restantes páginas do boletim é tomado como garantido que estas apresentam a mesma orientação da página inicial.



(a) Orientação correta



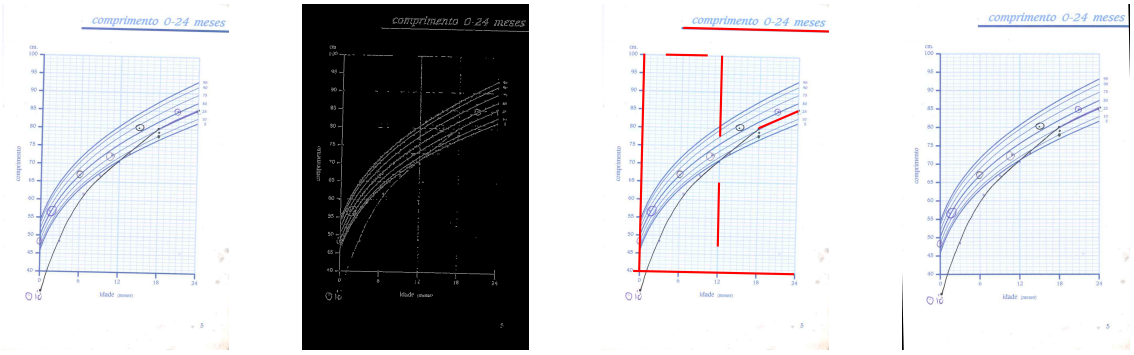
(b) Páginas invertidas

Figura 3.6: Detecção de bordas e de inversão de página aplicada à figura 3.4

Por outro lado, a correção de pequenas rotações das páginas é feita individualmente a cada página. Mais uma vez, é aplicada a máscara da cor do género. Esta é aplicada para extrair as linhas presentes na página, quer no cabeçalho da página, quer na tabela ou gráfico. Detectadas estas linhas é aplicada um novo método do OpenCv chamado ***HoughLinesP***, que consiste numa

implementação da transformada de *Hough*. A transformada de *Hough* é uma técnica matemática que realiza a detecção de formas geométricas em imagens digitais e, neste caso, é utilizada para a deteção de linhas rectas.

Obtidas as potenciais rectas, é calculado o declive das mesmas. Através dos declives é prevista a inclinação global da página, tendo em conta que as principais linhas nos boletins são estritamente verticais ou horizontais. Calculado o ângulo de rotação da página é aplicada uma rotação à mesma, com origem no ponto central da página. A figura 3.7 representa as diferente etapas deste processo.



(a) Original (b) Máscara de cor e detecção de bordar (c) Detecção linhas (d) Resultado final

Figura 3.7: Correcção de rotação

3.2.2 Detecção do género

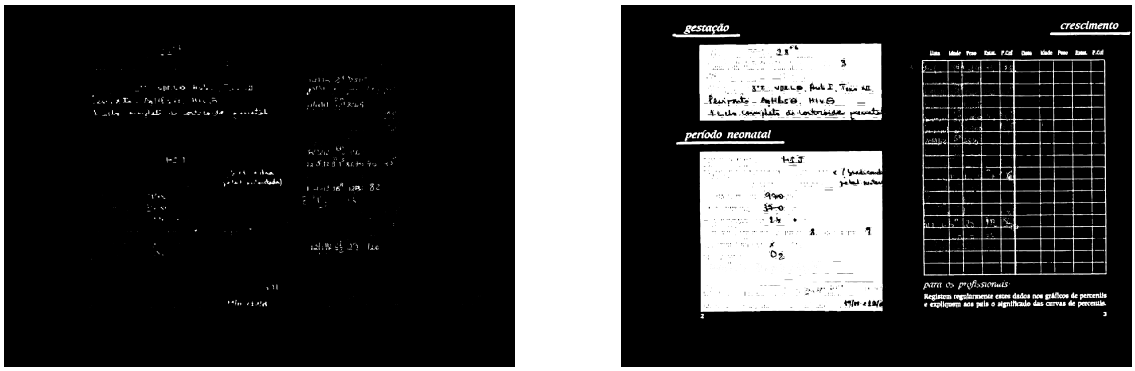
Os boletins de diferentes géneros são distinguíveis pela cor do conteúdo: azul nos rapazes e rosa nas raparigas. Assim sendo, analisando os boletins a maior densidade de cor é encontrada logo na primeira páginas dos PDFs, que corresponde à 2 página dos boletins. Esta página pode ser observada na figura 3.4.

É então procedida a criação de duas máscaras de cor azul e rosa. É utilizado o espaço de cor HSV e obtidas as matizes das cor azul e rosa. Neste espaço de cor, o primeiro canal (matiz) é representado por valores entre 0 e 360º e cada uma das 6 cores principais (vermelho, magenta, azul, ciano, verde e amarelo) tem um intervalo de 60º. Tendo em conta que o espaço HSV no OpenCV é comprimido para metade, o intervalo para cada cor é dado pela seguinte expressão:

$$Intervalo = [\frac{matiz - 30}{2}, \frac{matiz + 30}{2}] \tag{3.1}$$

Com o intervalo criado e a imagem alvo definida, as máscaras são aplicadas. Pode ser visualizado um exemplo na imagem 3.8.

Depois de aplicadas ambas as máscaras, são comparadas as imagens resultantes. Para esta comparação é contado o número de pixeis não nulos. A máscara com uma imagem resultante com maior número de pixeis não nulos representará o género da criança. No caso da imagem 3.8, a máscara usada em 3.8b teve um maior alcance, portanto, conclui-se que pertence a uma rapariga.



(a) Página com máscara azul

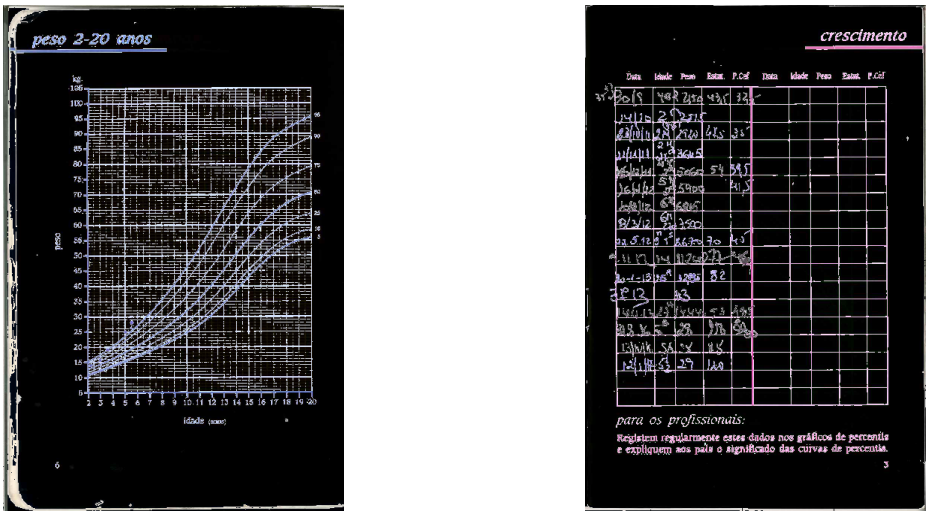
(b) Página com máscara rosa

Figura 3.8: Detecção do género, aplicada à imagem 3.4

3.2.3 Isolar conteúdo da página

A última etapa em comum no pré-processamento de páginas com gráfico e páginas com tabela é o isolamento do seu conteúdo. Este processo tem como objetivo abstrair toda a informação não essencial para que possa ser aplicado um processamento mais localizado.

Para obter esse resultado, primeiramente, como já foi feito anteriormente noutro processo, a imagem sofre uma redução no seu tamanho para optimizar a deteção de elementos, por exemplo, de contornos. De seguida é removido o fundo branco das páginas, de modo a aumentar o contraste do conteúdo. Esse feito é alcançado aplicando uma máscara que remove todos os pixeis com valores entre 225 e 223 em todos os canais do espaço de cor BGR. Recorrendo a várias tentativas foi este o intervalo que levou a um melhor isolamento da cor branca. Aplicada esta máscara, obtêm-se resultados idênticos aos que podem ser vistos na figura 3.9

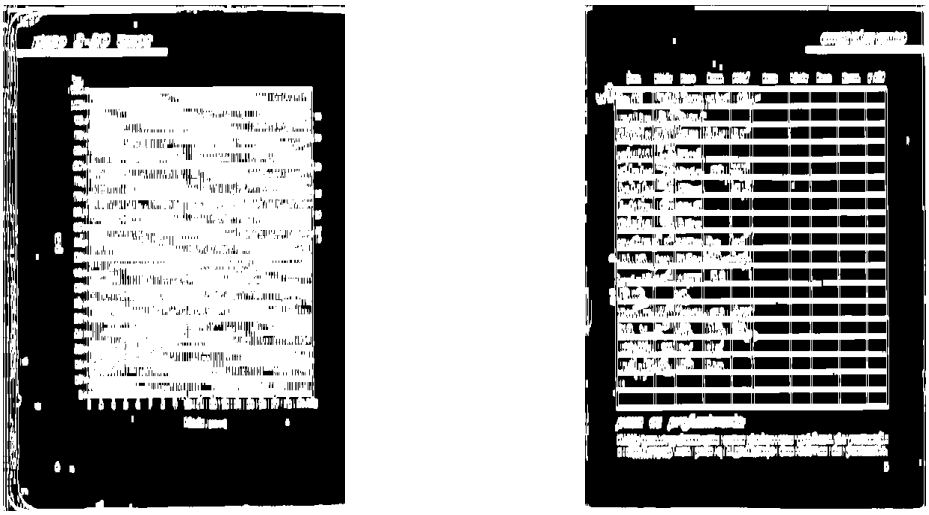


(a) Máscara de remoção do fundo branco aplicada a um gráfico

(b) Máscara de remoção do fundo branco aplicada a uma tabela

Figura 3.9: Remoção do fundo branco

Os passos seguintes neste processo passam por aplicar um filtro gaussiano para remover pequenas imperfeições e utilizar o método ***Canny*** do OpenCv, já referido anteriormente, para a deteção dos contornos do conteúdo. É importante assegurar a inclusão dos valores dos eixos dos gráficos e de qualquer dado das tabelas que se encontrem forem das células, como tal, são aplicadas transformações morfológicas de fechamento, tanto com um *kernel* horizontal como vertical, para "unir"esses valores externos ao gráfico/tabela. Para melhores resultados, é ainda aplicada uma transformação morfológica de dilatação aos contornos. Um exemplo dos resultados obtidos pode ser observado na figura 3.10



(a) Página de gráfico após realce do conteúdo

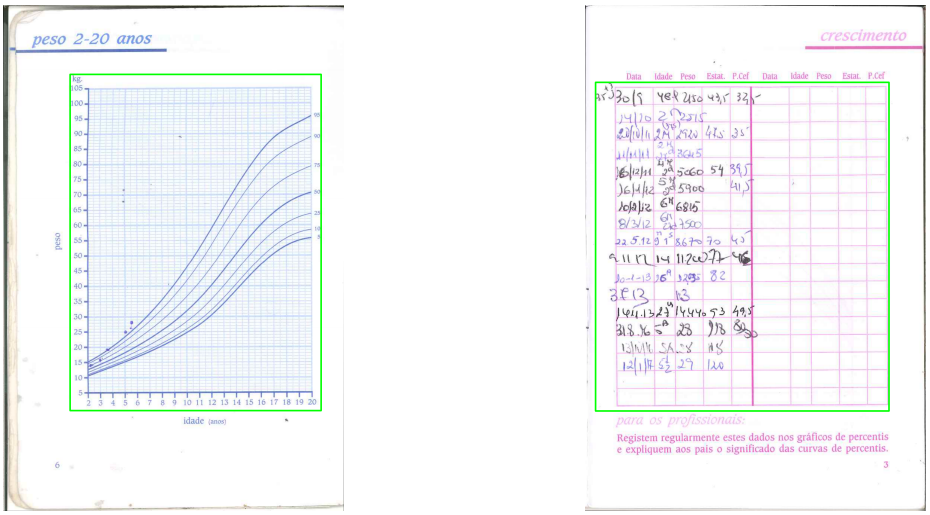
(b) Página de tabela após realce do conteúdo

Figura 3.10: Realce do conteúdo de uma página

Tendo já o conteúdo da página realçado, o passo seguinte passa por identificar os limites do conteúdo. Para tal, é chamado o método ***findContours*** do OpenCv para delimitar o conteúdo por um rectângulo. Esta deteção é restinguida pela condição que a área da tabela/gráfico está entre 40% e 70% da área total da figura. É tomado como correto o contorno com maior área que esteja entre os limites anteriores. Os contornos resultantes são idênticos aos que podem ser vistos na figura 3.11

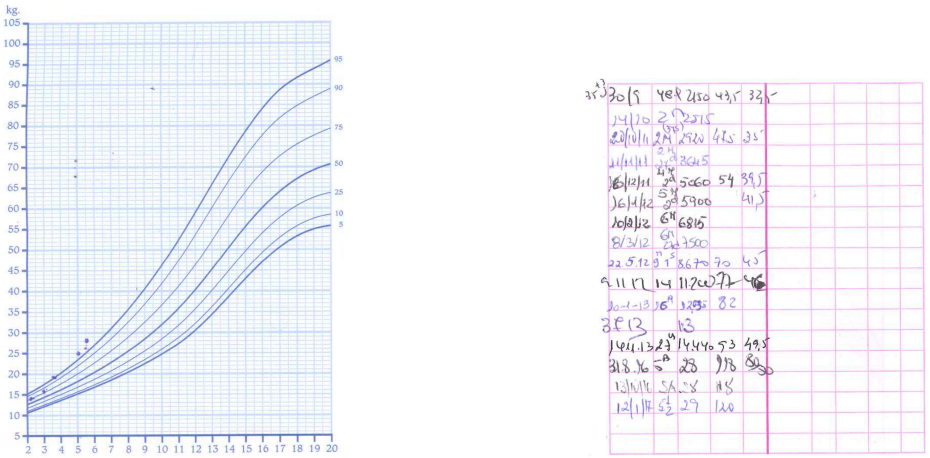
Por fim, são utilizadas as coordenadas dos contornos e extraído todo o conteúdo externo, para a maior abstracção possível nas etapas seguintes. Relembrando que no início deste processo de isolamento a imagem da página foi reduzida em tamanho. Agora, às coordenadas é aplicado o factor de redução para que estas correspondam à imagem original. É importante a utilização do tamanho original para não perder detalhe no conteúdo.

Atendendo aos contornos da figura 3.11, o resultado obtido seria o que pode ser observado na figura 3.17



(a) Detecção um gráfico (b) Detecção de uma tabela

Figura 3.11: Detecção do conteúdo de uma página



(a) Conteúdo de um gráfico (b) Conteúdo de uma tabela

Figura 3.12: Conteúdo de uma página

3.2.4 Páginas com gráfico

A partir deste ponto há uma bifurcação na *pipeline*, sendo que as página que contenham um gráfico receberão um processamento diferente daquelas que contenham uma tabela. Nos tópicos seguintes será abordada apenas a *pipeline* dos gráficos.

3.2.4.1 Identificar eixos

Na *pipeline*, neste ponto, já o gráfico já se encontra isolado de maneira semelhante à que pode ser vista na figura 3.12a.

Todas as linhas dos gráficos, quer os eixos quer as guias, são representadas pela cor do boletim, como já referido, azul nos boletins dos rapazes e rosa nos das raparigas. Assim sendo, é utilizado o resultado da detecção do género, explicada na secção 3.2.2, e aplicada uma máscara com a cor correspondente com a finalidade de isolar as linhas. Utilizando o gráfico da figura 3.12a como exemplo, o resultado seria a figura 3.13

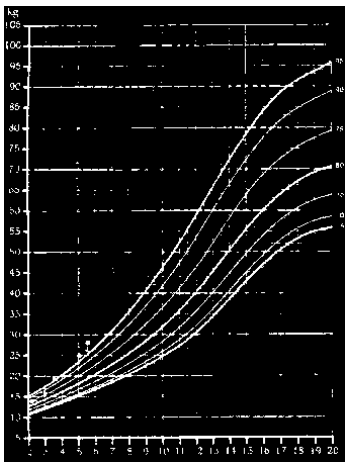


Figura 3.13: Resultado da máscara de cor azul aplicada ao gráfico 3.12a

O resultado do passo anterior pode apresentar linhas com ligeiras descontinuidades ou linhas mal definidas. Para minimizar essas situações são aplicadas transformações morfológicas de fechamento, tanto com um *kernel* horizontal como vertical, e de dilatação. Com as linhas melhores definidas é aplicado o método ***HoughLinesP*** do OpenCv, resultado em *n* linhas, tanto horizontais como verticais. Deste conjunto de resultados, o eixo das abcissas é a linha que:

- tenha uma inclinação com valor absoluto menor que 5 graus;
- se encontre a uma distância do limite inferior entre 2.5% e 7% da altura total da imagem

Por outro lado, o eixo das ordenadas é a linha que:

- tenha uma inclinação com valor entre 85 e 95 graus;
- se encontre a uma distância do limite esquerdo/3 entre 5% e 10% da largura total da imagem

Acrescendo a estas condições individuais, têm-se ainda que o ângulo formado por estas duas rectas têm de ser aproximadamente de 90º.

Obtidas as duas rectas que formam os eixos, é calculada a intersecção destas para obter a origem do gráfico. A figura 3.14 ilustra o resultado deste algoritmo aplicado ao gráfico da figura 3.12a.

3.2.4.2 Identificar escala dos eixos dos gráficos

Como os valores dos eixos dos gráficos são lineares apenas é necessário detectar o limite superior e inferior de cada um dos eixos, ou seja, 4 valores distintos. Esses valores podem ser encontrados:

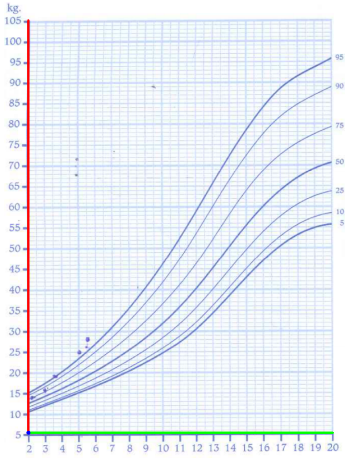


Figura 3.14: Detecção dos eixos do gráfico 3.12a

1. **Limite superior do eixo yy:** à esquerda do ponto superior do eixo yy
2. **Limite inferior do eixo yy:** à esquerda da origem do gráfico
3. **Limite inferior do eixo xx:** abaixo da origem do gráfico
4. **Limite superior do eixo xx:** abaixo do limite direito do eixo xx

Tendo as zonas definidas, o reconhecimento dos caracteres é feita recorrendo ao *Google's Tesseract OCR*. Para otimizar os resultados, o reconhecimento foi limitado aos 10 algarismos arábicos. Para detectar o valor em cada uma das zonas, a configuração utilizada foi a presente na listagem 2.

```
1 valor = pytesseract.image_to_string(zona, config='--psm 10 --oem 3 -c '
2                                             'tessedit_char_whitelist=0123456789')
```

Listing 2: Google's Tesseract OCR para a deteção dos valores dos eixos

Para aumentar a consistência dos valores e para prevenção de qualquer falha na leitura de algum dos valores nos limites dos eixos, é ainda obtida uma terceira leitura em cada eixo. Esse valor é o central de cada eixo e, em caso de falha numa das extremidades, possibilita a sua recuperação atendendo à linearidade dos valores dos eixos.

Na figura 3.15 são visíveis as áreas identificadas.

3.2.4.3 Identificar pontos

Com o objetivo de identificar os pontos dos gráficos, foram desenvolvidas duas implementações com bases diferentes e com resultados diferentes. Nos tópicos seguintes vai ser analisada cada uma das implementações e feita uma comparação entre elas.

Como exemplo, partiremos da imagem 3.16.

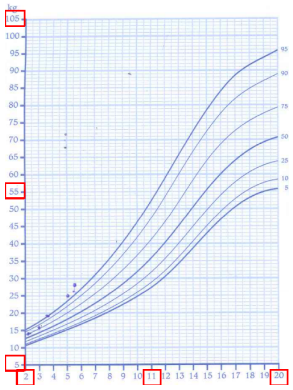


Figura 3.15: Detecção da escala dos eixos

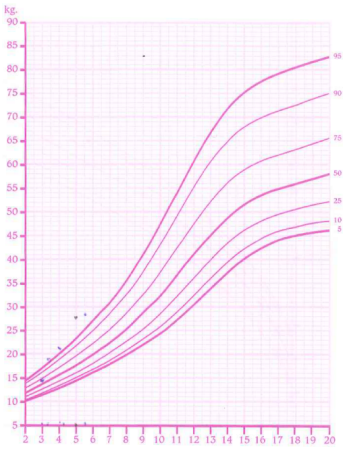


Figura 3.16: Detecção dos pontos: gráfico original

Em ambos os métodos, o gráfico original é comprimido numa escala 1:5 para otimizar os resultados pois, com a condensação dos pixeis, são excluídas pequenas irregularidades da vizinhança. No caso da detecção de pontos por *template matching*, leva ainda a que se obtenha correlações superiores entre *templates* e pontos, dado a área de comparação vai ser menor.

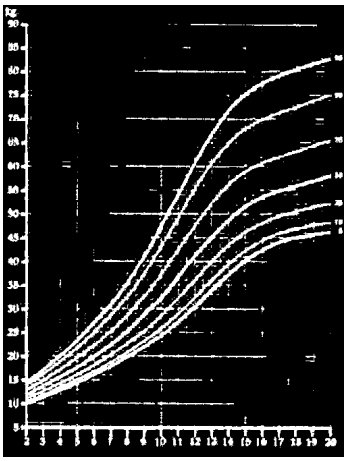
3.2.4.3.1 Detecção dos pontos por cor

Como o título indica, por este método, os pontos são detectados pela sua cor. Para começar, são aplicadas 2 máscaras distintas à imagem original:

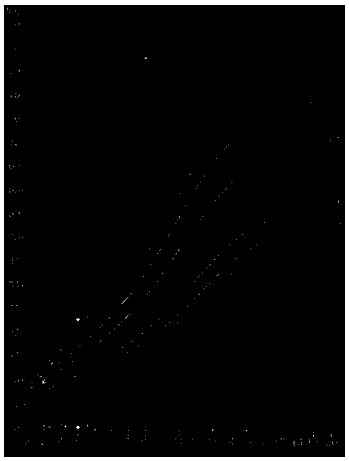
1. máscaras correspondente à cor do boletim (azul ou rosa) para detetar os eixos e a grelha do gráfico (figura 3.17a);

2. máscara para detetar posições mais escuras na imagem. Esta máscara corresponde aos pixeis com todos os valores do espaço de cor BGR menores que 190. O objetivo desta máscara é isolar a tinta da caneta, por normal azul escura ou preta (figura 3.17b).

Como pode ser visto na figura 3.17b, o resultado inclui os pontos e resíduos da gralha do gráfico. Assim, para um resultado mais limpo, é obtida um outra máscara a partir de:



(a) Detecção dos pontos por cor: máscara 1



(b) Detecção dos pontos por cor: máscara 2

Figura 3.17: Detecção dos pontos por cor: máscaras iniciais

$$Mascara = Mascara2 - Mascara1 \tag{3.2}$$

O resultado desta nova máscara, serão na teoria, apenas os pontos, como pode ser visto em 3.18.

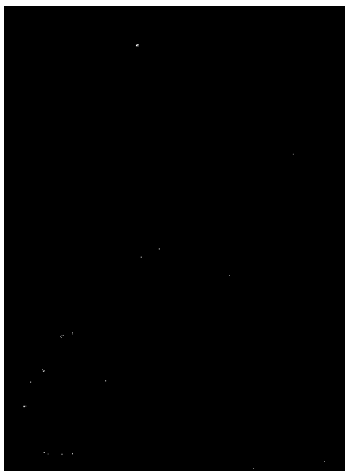


Figura 3.18: Detecção dos pontos por cor: máscara final

Para detetar a posição dos pontos, é utilizado o método ***findContours*** de OpenCv, ao qual é imposto, como única condição que este tenha uma área entre 10 e 200 pixeis.

O resultado obtido, neste caso, seria o apresentado na figura 3.19.

3.2.4.3.2 Detecção dos pontos por *template matching*

Ao contrário do método anterior, neste não é aplicada qualquer máscara. O único pré processamento aplicado é a passagem do gráfico para escalas de cinza.

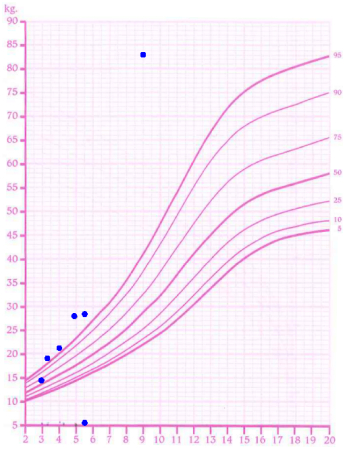


Figura 3.19: Detecção dos pontos por cor: resultado

Por outro lado, exige uma pré seleção de *templates* de pontos. Para esta execução, foi preparado apenas um pequeno conjuntos de pontos, como pode ser visto na figura 3.20. De salientar que a cor dos *templates* não tem qualquer interferência com o resultado final, visto, ao serem carregados, estes serem convertidos para escalas de cinza, tal como o gráfico.

É importante que estes *templates* cubram a maior variedade de formas de pontos possível, para otimização dos resultados obtidos.

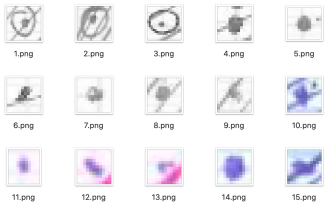


Figura 3.20: Detecção dos pontos por *template matching*: *templates*

Para efectuar o *template matching* foi utilizado o método ***matchTemplate*** do OpenCv. Este método apenas permite a entrada de um *template*, como tal, foi necessário fazer um ciclo para o utilizar para todos os pontos salvos como *templates*. Desta forma, é obtida uma matriz de correlações para cada *template*. A partir de cada matriz, são analisados os máximos locais, que na teoria correspondem aos pontos nos gráficos. Para um conjunto de pixeis ser considerado ponto, tem que passar pelas seguintes condições:

- ser máximo local numa das matrizes de resultados e ter uma correlação superior a 45% com um dos *templates*;
- haver pelo menos mais 1 matriz de resultados em que a correlação nesse ponto seja superior a 35%.

O resultado obtido, tendo também como entrada a imagem 3.16, é o que pode ser observado em 3.21.

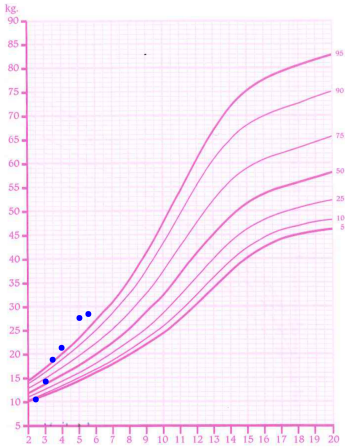
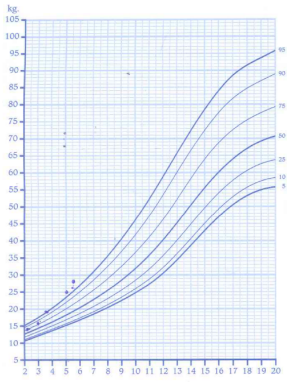


Figura 3.21: Detecção dos pontos por *template matching*: resultado

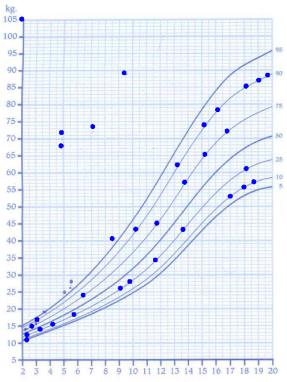
3.2.4.3.3 Comparação entre algoritmos

Analisando os resultados apresentados nas figuras 3.19 e 3.21, ambos parecem apresentar excelentes resultados. Porém, estes resultados são enganadores. Foi escolhido um caso óptimo para a execução da deteção por cor.

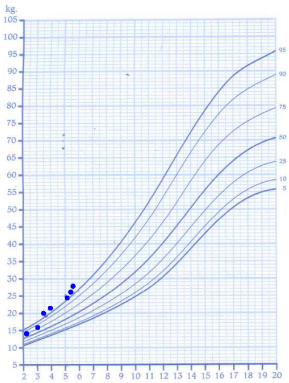
O cenário ideal para a execução da deteção de pontos por cor é em gráficos com grelha cor de rosa e pontos assinalados a caneta azul ou preta, que é o caso da situação exemplo apresentada. Nestas situações, este algoritmo apresenta óptimos resultados. Porém agora vamos analisar a execução de ambos algoritmos num cenário em que a grelha do gráfico é azul e os pontos estão também assinalados a caneta azul.



(a) Gráfico original



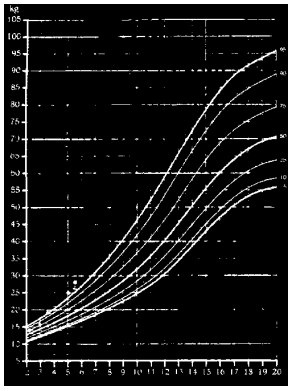
(b) Identificação de pontos por cor



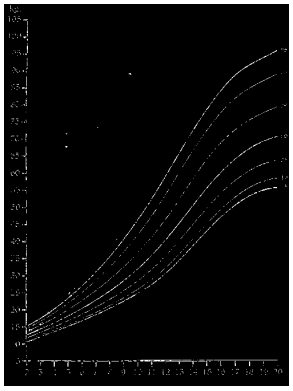
(c) Identificação de pontos por *template matching*

Figura 3.22: Detecção dos pontos: boletim azul

Na figura 3.22 é possível observar uma diferença abrupta entre os 2 algoritmos. Os maus resultados da deteção de pontos por cor deve-se à dificuldade de isolar a grelha azul dos pontos, também azuis. Essa dificuldade é visível através das análise das máscaras obtidas por este método na figura 3.23.



(a) Detecção dos pontos por cor: máscara 1



(b) Detecção dos pontos por cor: máscara 2



(c) Detecção dos pontos por cor: máscara resultante

Figura 3.23: Detecção dos pontos por cor: máscaras obtidas

Analisando a máscara 1 (que é uma máscara para a cor azul, visível na figura 3.23a), denota-se a inclusão tanto da grelha da tabela como dos pontos. Atendendo a que:

$$MascaraRes = Mascara2 - Mascara1 \tag{3.3}$$

se os pontos tiverem inclusos nas máscara 1, estes não estarão presente na máscara resultante, que será a máscara analisada para a deteção dos pontos.

Para o algoritmo de identificação de pontos por cor ser bem sucedido, os pontos e a grelha dos gráficos, no espaço de cor HSV, têm de apresentar matizes com valores amplamente distintos.

Por outro lado, o algoritmo de identificação de pontos por *template matching* apresenta bons resultados em qualquer situação de cor, devido ao facto deste se basear apenas na forma dos pontos.

3.2.4.3.4 Conclusões

Como já foi analisado, o algoritmo de identificação de pontos por *template matching* obtém melhores resultados em ambos os casos, boletins azuis e boletins cor de rosa. Apesar de necessitar de uma preparação prévia de *templates* de pontos (também pode ser uma vantagem pois é possível adicionar novos *templates* se necessário incluir novas formas de pontos) e exigir um maior esforço computacional, a estas desvantagens é sobreposta a consistência que apresenta e que é um factor importante no âmbito desta dissertação. Assim, na *pipeline*, o método utilizado é o de identificação de pontos por *template matching*.

3.2.4.4 Eliminação de *outliers*

De modo a não perder informação importante neste passo, são eliminados valores completamente desajustados com o resto do conjunto. São considerados *outliers* quando:

- representam uma descida em gráficos como o comprimento e o perímetro cefálico;

- apresentam uma subida/descida acentuada, formando um ângulo de mais de 75% com o ponto anterior.

3.2.4.5 Relacionar pontos com eixos

Neste ponto já são conhecidos tanto os valores dos eixos como as coordenadas dos pontos, coordenadas estas já relacionadas com a posição do eixo do gráfico (a coordenada (0,0) encontra-se sobre o limite superior do eixo dos xx e a origem do gráfico encontra-se na coordenada (0, altura da imagem)). Assim, a conversão é quase direta, aplicando a escala dos gráficos, detectada anteriormente.

3.2.5 Páginas com tabela

Nos tópicos anteriores foi referida a *pipeline* para as páginas do boletim que continham gráfico. Agora, serão analisados os processos a aplicar em caso do conteúdo ser uma tabela.

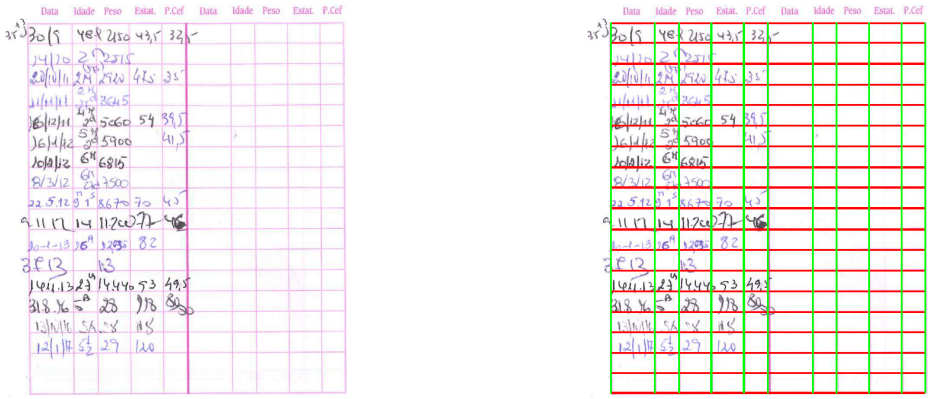
3.2.5.1 Detectar grelha da tabela

Partindo o resultado obtido em 3.2.3, o passo seguinte na *pipeline* de uma tabela consiste na deteção das linhas que formam a sua grelha.

O pré-processamento aplicado para a deteção e enaltecimento é o mesmo aplicado em 3.2.4.1, na identificação dos eixos dos gráficos. Como tal, são aplicadas uma máscara da cor correspondente ao boletim para isolar as linhas e transformações morfológicas de fechamento e de dilatação para melhorar a estrutura das linhas. Por fim, para obter as linhas é utilizado o algoritmo ***HoughLinesP*** do OpenCv.

Em casos que as tabelas estão com elevado nível de preenchimento, é possível que os passos anteriores resultem na obtenção de linhas que não correspondem às da grelha da tabela e na não obtenção de algumas linhas que definitivamente correspondem à tabela. Assim sendo, e por razões de consistência, foi desenvolvido um algoritmo para relacionar as linhas obtidas com a grelha da tabela e, em caso da falha na deteção de alguma das linhas, proceder ao restauro da mesma no local onde seria mais provável se encontrar. O algoritmo varia ligeiramente das linhas verticais para as linhas horizontais. O algoritmo base é aplicado nas linhas horizontais, visto que o espaçamento entre linhas é constante. Assim, tendo procedido à deteção das linhas horizontais, é analisado o espaçamento entre elas. Através dos espaçamentos entre as linhas é construída uma cadeia de possíveis localizações das linhas. O espaçamento e as posições reais das linhas da tabela serão aqueles que, com base nas linhas obtidas no processo anterior, obtenham a cadeia de tamanho mais próximo ao real (as tabelas dos boletins têm 19 linhas horizontais) e com maior correlação entre a posição das linhas e o espaçamento obtido. No caso das linhas verticais, o processo difere ligeiramente pois o espaçamento não é constante. Por outro lado, temos que existe uma linha vertical central que divide a tabela em 2 partes iguais, com espaçamentos iguais. Deste modo, o conjunto de linhas verticais a ser considerado é aquele que tiver melhor correlação de posições e espaçamento entre os dois lados da tabela.

Na figura 3.24 é possível observar tanto a deteção das linhas como a execução do algoritmo implementado. Para forçar a falha na deteção de uma das linhas, foi removida a penúltima linha horizontal da tabela, recorrendo ao Photoshop (ver figura 3.24a). Na figura 3.24b é apresentada a deteção com sucesso de toda a grelha da tabela, incluindo a linha retirada propositadamente.



(a) Tabela original

(b) Conteúdo de uma tabela

Figura 3.24: Detecção da grelha de uma tabela

3.2.5.2 Isolar células

Partindo da grelha da tabela, são facilmente identificadas as linhas e as colunas da mesma. Com duas linhas verticais e duas linhas horizontais consecutivas é possível obter os 4 pontos que formam uma célula da tabela. (figura 3.25)

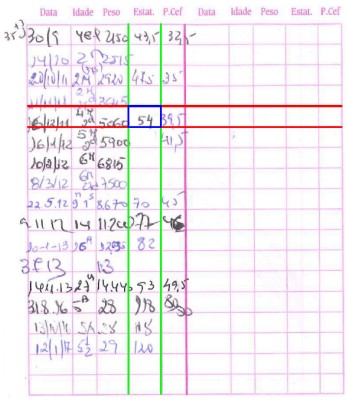


Figura 3.25: Isolamento de célula da tabela

3.2.5.3 Ler conteúdo das células

De forma a otimizar o reconhecimento do conteúdo das células é feito o isolamento dos caracteres. Para salientar os caracteres são aplicadas as seguintes técnicas: conversão da imagem para escala de cinzentos, aplicar filtro de gaussiano para remover pequenas imperfeições, aplicar um

threshold para isolar a tinta da caneta e aplicar transformações morfológicas de fechamento e dilatação para restaurar e enaltecer os caracteres. Para detectar os limites dos caracteres é utilizado o método ***findContours*** do OpenCv. São considerados caracteres válidos, aqueles que tiverem uma área entre 2 e 40% da área total da célula.



Figura 3.26: Detecção de caracteres

Posteriormente, é feito o reconhecimento ótico de cada caractere individual. Em caso de sucesso é salvo o valor, em caso de falha é sinalizada a presença do caractere para ser possível determinar o intervalo em que o valor real se encontra.

Foi utilizada mais que uma ferramenta para o reconhecimento dos caracteres e comparada classificação das diferentes ferramentas para obter um resultado mais preciso.

As ferramentas utilizadas serão apresentadas de seguida.

3.2.5.3.1 *Google Tesseract*

Foi utilizada a biblioteca *pyteserract* que é um *wrapper* para o *Google's Tesseract-OCR Engine*.

Apesar de apresentar melhores resultados para dígitos digitais, também pode ser utilizado para o reconhecimento de dígitos manuscritos. As grandes vantagens desta ferramenta é a capacidade de receber como entrada uma imagem de dimensões variáveis e de reconhecer múltiplos dígitos na mesma imagem. Este ponto pode ser útil em casos de má segmentação no dígito de entrada.

Para otimizar os resultados, o output foi limitado aos 10 algarismos da numeração árabe.

3.2.5.3.2 *Keras*

Foi utilizada a biblioteca Keras para treinar uma CNN para a classificação de dígitos manuscritos, recorrendo à base de dados **MNIST**.

O modelo desenvolvido é o apresentado na listagem 3.

Comparativamente com a ferramenta descrita anteriormente, as desvantagens desta implementação é a necessidade de formatar o input, sendo este somente um dígito branco centrado numa imagem de fundo preto, de dimensões *28x28*. Por outro lado, como output apresenta a correlação do input e os diferentes dígitos, não apenas do que tem maior correlação.

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D, Dropout, Flatten, MaxPooling2D

(x_train, y_train), (x_test, y_test) = tf.keras.datasets.mnist.load_data()
x_train = x_train.reshape(x_train.shape[0], 28, 28, 1)
x_test = x_test.reshape(x_test.shape[0], 28, 28, 1)
input_shape = (28, 28, 1)

x_train = x_train.astype('float32')
x_test = x_test.astype('float32')

x_train /= 255
x_test /= 255

model = Sequential()
model.add(Conv2D(28, kernel_size=(3, 3), input_shape=input_shape))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(128, activation=tf.nn.relu))
model.add(Dropout(0.2))
model.add(Dense(10, activation=tf.nn.softmax))

model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
model.fit(x=x_train, y=y_train, epochs=10)

model.evaluate(x_test, y_test)

model.save('mnist.model')
```

Listing 3: CNN para a classificação de dígitos manuscritos

3.2.6 Cruzar dados de ambas as fontes

Como caso padrão, de cada boletim, resultam 1 tabela e 6 gráficos. Relacionando as páginas do boletim com o seu conteúdo temos:

3. Tabela (com todas as informações de peso, comprimento e perímetro cefálico)
4. Gráfico: Peso 0-24 meses
5. Gráfico: Comprimento 0-24 meses
6. Gráfico: Peso 2-20 anos
7. Gráfico: Comprimento 2-20 anos

8. Gráfico: Perímetro cefálico 0-36 meses

9. Gráfico: Índice de massa corporal 2-20 anos

Assim sendo, e por razões de comparação, os valores que utilizam como escala os anos são convertidos para meses. Feito isto, os dados das páginas 4 e 6 e das páginas 5 e 7 podem ser unidos.

Como resultado, temos 4 conjuntos de dados diferentes: peso, comprimento, perímetro cefálico e índice de massa corporal. Os 3 primeiros correspondem a colunas na tabela, já o índice de massa corporal é útil para validar o comprimento e o peso.

$$IMC = \frac{Peso}{Altura^2} \tag{3.4}$$

Para proceder ao cruzamento dos dados é percorrida linha a linha da tabela. Cada linha tem o registo da idade da criança na altura das medições. Para cada medição, é acedido ao seu valor na tabela e procurado, nos gráficos correspondentes, os valores obtidos para aquela idade (é dada uma tolerância de 1 mês para os gráficos dos 0-24 meses e de 6 meses para os gráficos dos 2-20 anos). Se encontrar um valor para aquela idade e esse valor for igual ou com uma ligeira margem de erro face ao valor da tabela, a esse valor é dada uma *flag* de elevado grau de certeza. Se não for encontrado um valor para aquela idade, é atribuída ao valor uma *flag* de nível de certeza intermédio. Relembrando também os casos de falha de leitura no valor da tabela, mas que é possível determinar um intervalo para o valor, se for possível encontrar correspondência nos gráficos, a esse valor também será atribuída uma *flag* de nível de certeza intermédio. Com *flag* de baixo nível de certeza temos os casos de valores dos gráficos que não tem qualquer correspondência com a tabela e os dados da tabela que tiveram falhas na leitura e não representam nenhum ponto nos gráficos no seu intervalo (nestes casos, o valor salvo é o valor médio do intervalo).

3.2.7 Exportação dos dados

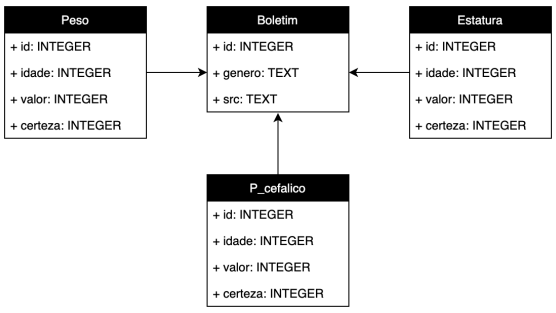


Figura 3.27: Estrutura da base de dados

Na figura 3.27 é apresentada a estrutura da base de dados. Nesta fase, a base de dados foi implementada em SQLite, mas é facilmente escalável para outro tipo se a quantidade de dados assim o necessitar.

Fica apresentada de seguida, no *listing* 4, a implementação da base de dados em SQLite.

```
create table boletim(
    id INTEGER PRIMARY KEY,
    genero TEXT NOT NULL,
    src TEXT NOT NULL
);

create table peso(
    id INTEGER PRIMARY KEY,
    boletim INTEGER,
    idade INTEGER NOT NULL,
    valor INTEGER NOT NULL,
    certeza INTEGER NOT NULL,
    FOREIGN KEY(boletim) REFERENCES boletim(id)
);

create table estatura(
    id INTEGER PRIMARY KEY,
    boletim INTEGER,
    idade INTEGER NOT NULL,
    valor INTEGER NOT NULL,
    certeza INTEGER NOT NULL,
    FOREIGN KEY(boletim) REFERENCES boletim(id)
);

create table p_cefalico(
    id INTEGER PRIMARY KEY,
    boletim INTEGER,
    idade INTEGER NOT NULL,
    valor INTEGER NOT NULL,
    certeza INTEGER NOT NULL,
    FOREIGN KEY(boletim) REFERENCES boletim(id)
);
```

Listing 4: Estrutura da base de dados em SQLite

Muito resumidamente, a tabela **boletim** guarda a informação do **id** (obtido do nome do PDF), o **género** da criança detentora do boletim e a pasta onde estão salvos os passos da execução (**src**).

As tabelas **peso**, **estatura** e **p_cefalico** guardam informações sobre o **boletim** de origem, a **idade** em que foram apontados, o **valor** registado e, por fim, a **certeza** do valor, explicada anteriormente.

Capítulo 4

Casos de estudo

Neste capítulo serão analisados os resultados obtidos para diferentes situações. Para tal, terá duas principais divisões: boletins cor de rosa e boletins azuis. Dentro destes ainda haverá uma divisão de casos fáceis e casos difíceis.

Quanto à deteção de pontos, entende-se por "caso fácil"quando está presente uma cadeia singular de pontos bem definidos e por "caso difícil"quando existem múltiplas sequências de pontos ou pontos mal definidos.

Quanto à extracção de informação das tabelas, entende-se por "caso fácil"quando o preenchimento das células está bem definido, dentro dos limites e com os diferentes dígitos bem segmentados. Um "caso difícil"é quando não cumpre os requisitos referidos.

4.1 Detecção de Pontos

Demonstração e comparação dos resultados obtidos em cenários de dificuldade diferente quanto à deteção de pontos em boletins azul e rosa.

4.1.1 Boletins azuis

Para este caso serão utilizadas páginas extraídas de um boletim de rapaz e como tal, os destaques das páginas são de cor azul.

4.1.1.1 Situação fácil

Para esta situação, foi escolhida a página 6 do boletim com ID 1026. Este apresenta apenas 6 pontos razoavelmente bem definidos e alguns *outliers* na parte superior, facilmente eliminados pelo seu desfasamento com o restante conjunto de pontos.

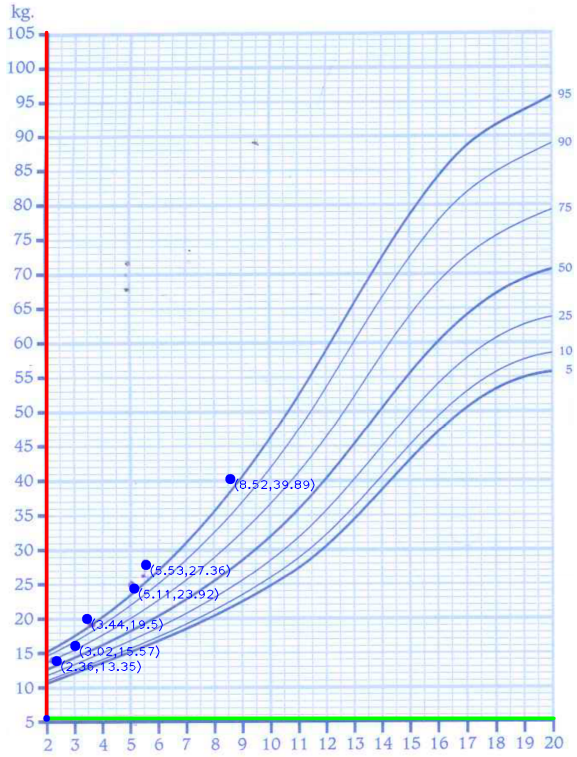


Figura 4.1: Boletim azul: deteção de pontos - caso fácil

Dos 6 pontos presentes no gráfico, 5 deles foram detectados com sucesso. Para além destes, foi considerado como ponto um *outlier*.

Tabela 4.1: Boletim azul: deteção de pontos - caso fácil (resultados)

nº pontos	**nº pontos detectados**	**nº *outliers***	**% de acerto**
6	5	1	83%

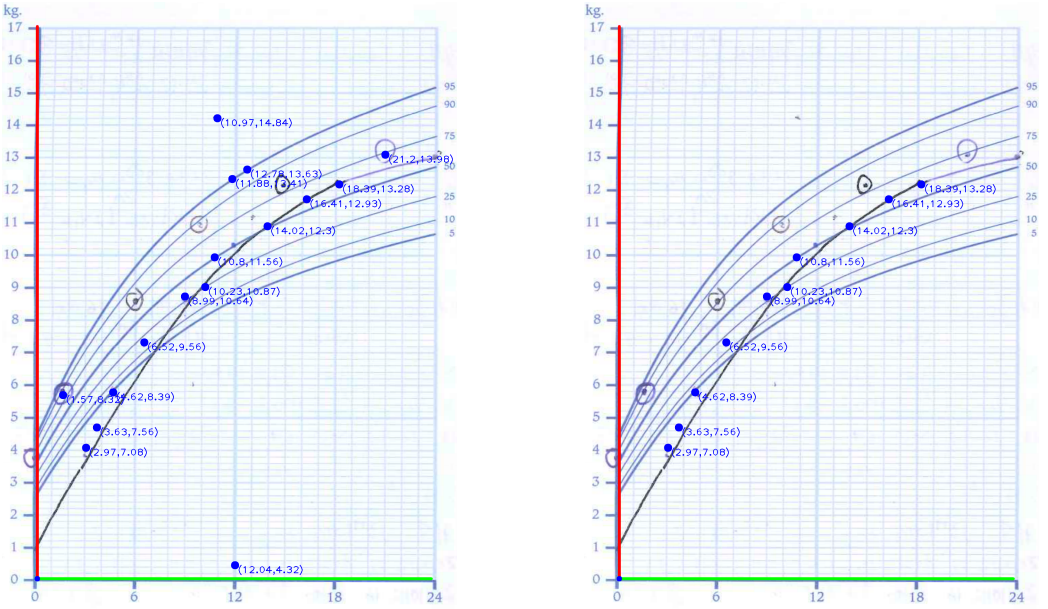
Comparando os valores obtidos com os valores reais (analisados pelo gráfico e pela tabela correspondente), obtêm-se os desvios apresentados na tabela 4.2. Não é possível medir com exactidão a precisão do algoritmo, tendo em conta que os desvios são causados não só for erros na deteção dos pontos mas também erros na sua marcação.

Tabela 4.2: Boletim azul: deteção de pontos - caso fácil (desvios)

Valores Reais	**Valores Obtidos**	**Desvios %**
(2.10 , 13.00)	(2.36 , 13.35)	(12.4% , 2.7%)
(3.00 , 15.80)	(3.02 , 15.57)	(0% , 1.5%)
(3.50 , 19.20)	(3.44 , 19.50)	(1.7% , 1.6%)
(5.15 , 23.6)	(5.11 , 23.92)	(0.7% , 1.4%)
(5.36 , 25.5)	-	-
(5.50 , 27.00)	(5.53 , 27.36)	(0.5% , 1.1%)
-	(8.52 , 39.89)	-

4.1.1.2 Situação difícil

Para esta situação, foi escolhida a página 4 do boletim com ID 1026. O gráfico desta página apresenta múltiplas cadeias de pontos, não sendo possível determinar a correta sem analisar a tabela.



(a) Detecção de pontos

(b) Detecção de pontos + algoritmo de exclusão de cadeia

Figura 4.2: Boletim azul: deteção de pontos - caso difícil

Nas imagens da figura 4.2 é possível analisar que nestes casos a deteção de pontos é menos eficaz. Em primeiro lugar, os pontos rodeados apresentam fraca eficácia no reconhecimento (com novos *templates* para estes pontos a eficácia pode vir a melhorar). Em segundo, foi desenvolvido um algoritmo para tentar eliminar uma das cadeias de pontos, o que por vezes leva a eliminar a cadeira errada (figura 4.2b). Assim, nestes casos são salvos todos os pontos para posterior comparação com os dados obtidos na tabela do boletim.

Tabela 4.3: Boletim azul: deteção de pontos - caso difícil (resultados)

nº pontos	nº pontos detectados	nº *outliers*	% de acerto
15	9	7	60%

4.1.2 Boletins rosa

Para este caso serão utilizadas páginas extraídas de um boletim de rapariga e como tal, os destaques das páginas são de cor rosa.

4.1.2.1 Situação fácil

Para esta situação, foi escolhida a página 6 do boletim com ID 1118. Este apresenta apenas 5 pontos bem definidos.

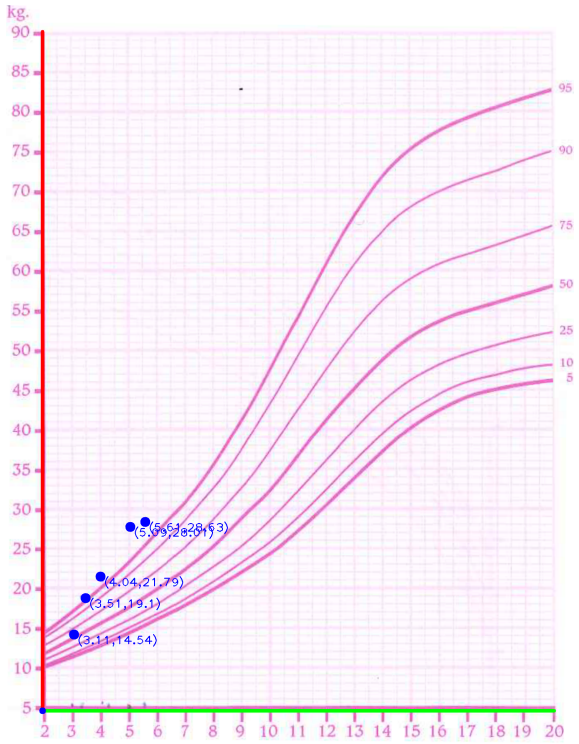


Figura 4.3: Boletim rosa: deteção de pontos - caso fácil

Neste caso, todos os pontos foram detectados com sucesso, sem a a inclusão de *outliers*. Representa, portanto, uma execução com uma percentagem de acerto de 100%.

Tabela 4.4: Boletim rosa: deteção de pontos - caso fácil (resultados)

nº pontos	**nº pontos detectados**	**nº *outliers***	**% de acerto**
5	5	0	100%

Fazendo a comparação dos valores obtidos com os valores reais, obtêm-se os desvios apresentados na tabela 4.5.

Tabela 4.5: Boletim rosa: deteção de pontos - caso fácil (desvios)

Valores Reais	**Valores Obtidos**	**Desvios %**
(3.00 , 14.40)	(3.11 , 14.54)	(3.7% , 0.9%)
(3.40, 19.00)	(3.51 , 19.10)	(3.3% , 0.5%)
(4.15 , 21.00)	(4.04 , 21.79)	(2.7% , 3.8%)
(5.00 , 28.00)	(5.09 , 28.01)	(1.18% , 0.0%)
(5.50 , 29.00)	(5.61 , 28.63)	(2.0% , 1.2%

4.1.2.2 Situação difícil

Para esta situação, foi escolhida a página 5 do boletim com ID 1118. O gráfico desta página apresenta múltiplas cadeias de pontos, não sendo possível determinar a correta sem analisar a tabela. Apresenta também uma linha que liga diversos pontos e dificulta na identificação destes.

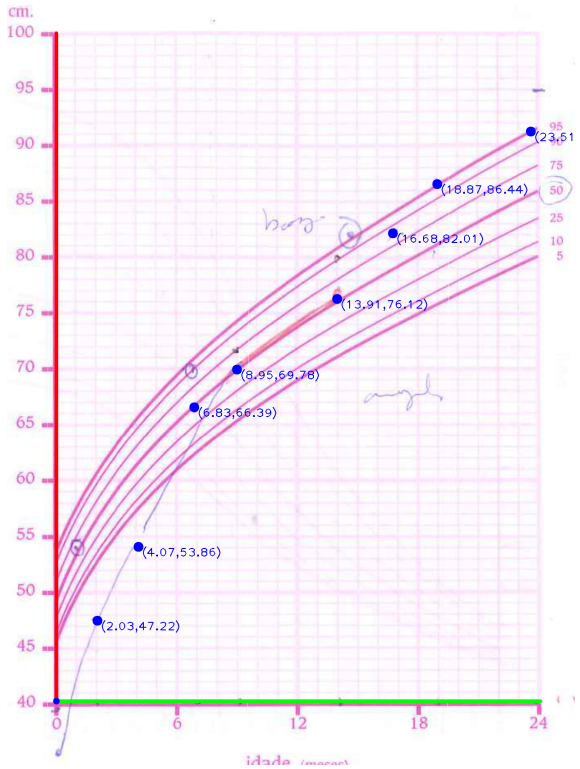


Figura 4.4: Boletim rosa: deteção de pontos - caso difícil

Mais uma vez, é visível na imagem 4.4 a dificuldade na deteção de pontos rodeados, detectando com sucesso os restantes, quase na totalidade. De salientar neste caso a não inclusão de nenhum *outlier*.

Tabela 4.6: Boletim rosa: deteção de pontos - caso fácil (resultados)

nº pontos	nº pontos detectados	nº *outliers*	% de acerto
12	8	0	67%

4.1.3 Conclusões

Analisando os resultados obtidos, é visível uma grande diferença no desempenho entre casos fáceis e casos difíceis, porém, esta diferença é mais relacionada com o fraco suporte para pontos rodeados. Estes representam 50% dos pontos não identificados em ambos os casos difíceis.

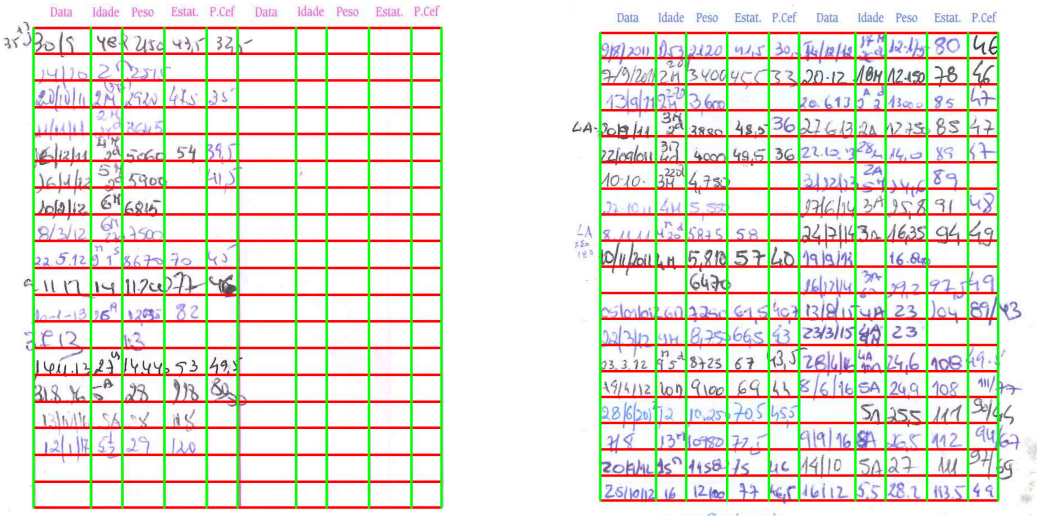
É também possível analisar uma superior percentagem de acerto, embora que ligeira, nos boletins cor de rosa. Esta vantagem deve-se ao maior contraste entre os pontos e a grelha e linhas dos gráficos, levando a uma maior facilidade em delimitar os pontos.

4.2 Extracção de informação das tabelas

Demonstração e comparação dos resultados obtidos em cenários de dificuldade diferente quanto à extracção de informação das tabelas em boletins azul e rosa. Esta fase da execução não apresenta variações de desempenho quanto à cor do boletim, apenas quanto ao nível de preenchimento das células da tabela.

4.2.1 Identificação das células

Foram seleccionadas duas tabelas, com diferentes níveis de preenchimento, para poder medir o desempenho de deteção das linhas da grelha das tabelas e isolar as células.



(a) Tabela pouco preenchida

(b) Tabela muito preenchida

Figura 4.5: Identificação das células da tabela - casos fácil e difícil

Através da 4.5 é possível concluir que o a identificação da grelha da tabela é bem sucedida em qualquer circunstância de preenchimento da tabela. Isto deve-se à implementação do algoritmo apresentado em 3.2.5.1, que detecta e recupera linhas em falta na grelha da tabela.

4.2.2 Segmentação e classificação dos caracteres

O sucesso da segmentação e classificação dos caracteres depende do grau de preenchimento da célula da tabela.

4.2.2.1 Caso fácil

Como caso fácil, temos um célula pouco preenchida e com os caracteres bem isolados. A célula seleccionada para este caso foi a apresentada na imagem 4.6.

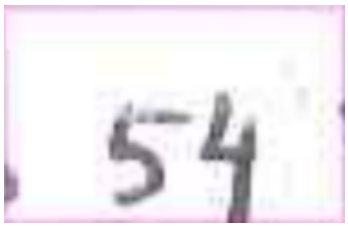
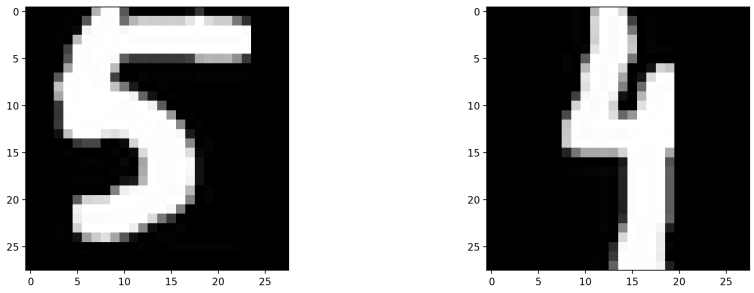


Figura 4.6: Segmentação de caracteres - caso fácil

Depois de aplicado o pré-processamento a esta célula é realizada a segmentação dos caracteres. Neste caso, o resultado desta operação é o apresentado na figura 4.7.



(a) Primeiro caractere (b) Segundo caractere

Figura 4.7: Caracteres resultantes de segmentação - caso fácil

Ambos os classificadores atribuiriam com sucesso a classificação de 5 ao primeiro caractere e de 4 ao segundo caractere. Unindo a informação das classificações de ambos os caracteres, conclui-se que o valor da célula é 54.

4.2.2.2 Caso difícil

Como caso difícil, temos uma célula com os dígitos sem separação entre si. A célula seleccionada para este caso foi a apresentada na imagem 4.8.

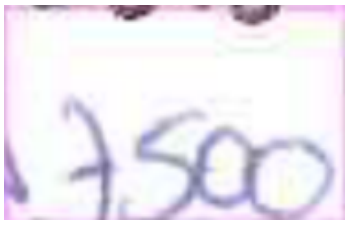
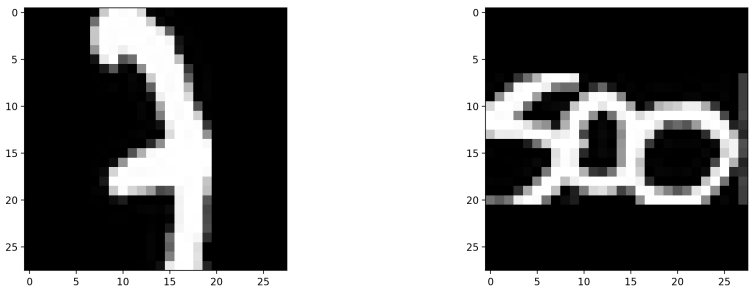


Figura 4.8: Segmentação de caracteres - caso difícil

Depois de aplicado o pré-processamento a esta célula é realizada a segmentação dos caracteres. Neste caso, o resultado desta operação é o apresentado na figura 4.9. Como os algarismos "500"está em contacto entre si, na segmentação dos mesmo, são inicialmente considerados como sendo um único caractere.



(a) Primeiro caractere (b) Segundo caractere

Figura 4.9: Caracteres resultantes de segmentação - caso difícil

Nesta situação, temos que um classificador atribui 3 ou outro 7 ao primeiro caractere. Ao segundo "caractere"nenhum dos classificadores conseguiu atribuir uma classificação válida. Porém, através do rácio altura-largura da área ocupada por este, é possível calcular que este é ocupado por 2 ou 3 dígitos. Assim temos que estes têm um valor no intervalo 00 - 999. Juntando os dois resultados da segmentação temos que o valor da célula se situa no intervalo 300 a 7999.

4.2.2.3 Conclusões

Apesar de terem sido implementadas medidas de contenção em caso de má segmentação, pode-se observar um enorme contraste entre os dois casos de dificuldade. Com uma boa segmentação pode atingir-se um resultado exacto, enquanto que em casos de má segmentação pode atingir-se, no melhor dos casos, um intervalo de valores, que pode ir de limitado a bastante vago.

Capítulo 5

Interface de utilizador

O foco desta dissertação é extrair os dados dos BSIJ, de modo a que estes sejam de fácil acesso.

Tendo já sido desenvolvida e analisada a exportação destes dados, a próxima etapa passa pela elaboração de uma interface que simplifique quer a importação de novos dados quer a validação e análise dos resultados obtidos. Esta interface deve ser focada nos seus potenciais utilizadores e nas suas necessidades.

5.1 Utilizadores

Atendendo à protecção e segurança destes dados, esta plataforma é reservada a médicos. Deste modo os utilizadores da aplicação podem ser divididos em:

- **Visitantes:** utilizadores não autenticados. Não têm acesso a dados;
- **Utilizadores autenticados:** utilizadores registados e autenticados. Têm acesso aos dados;
- **Administradores:** utilizadores autenticados, com permissões extra. Têm acesso aos dados e são responsáveis pela validação de novos utilizadores;

5.2 Funcionalidades

Nesta secção serão abordadas as diferentes funcionalidades e, em alguns casos, apresentado um exemplo visual.

As funcionalidades podem ser divididas pelos diferentes utilizadores da plataforma.

5.2.1 Visitantes

5.2.1.1 Login

Como visitante, o utilizador deve ser capaz de se autenticar, de modo a poder aceder aos dados.

Para efectuar o login, o visitante necessita do seu email de registo e da sua password.

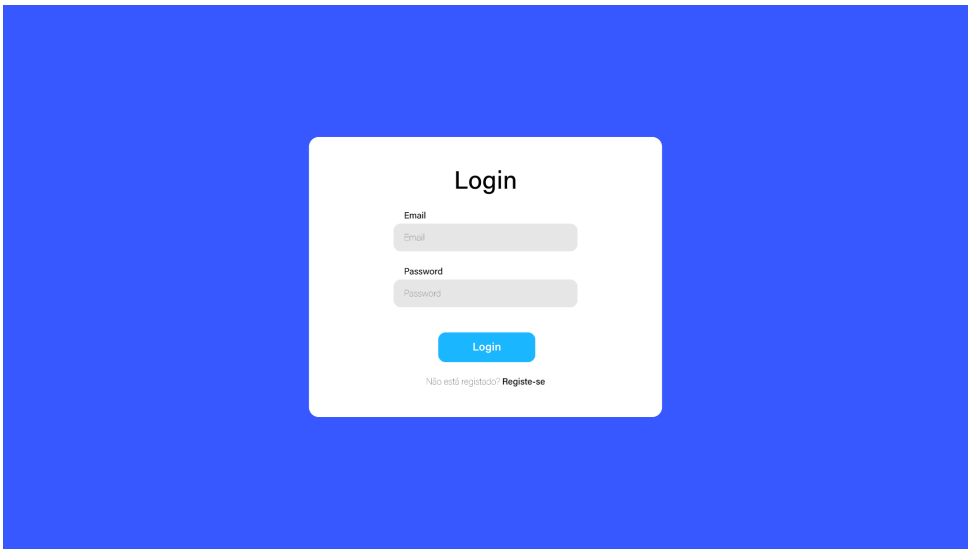


Figura 5.1: Mockup: Login

5.2.1.2 Registo

Como visitante, o utilizador, na ausência de credenciais de login, pode enviar um pedido de registo. Este será posteriormente validado por um administrador e, em caso de sucesso, serão concedidas permissões de acesso.

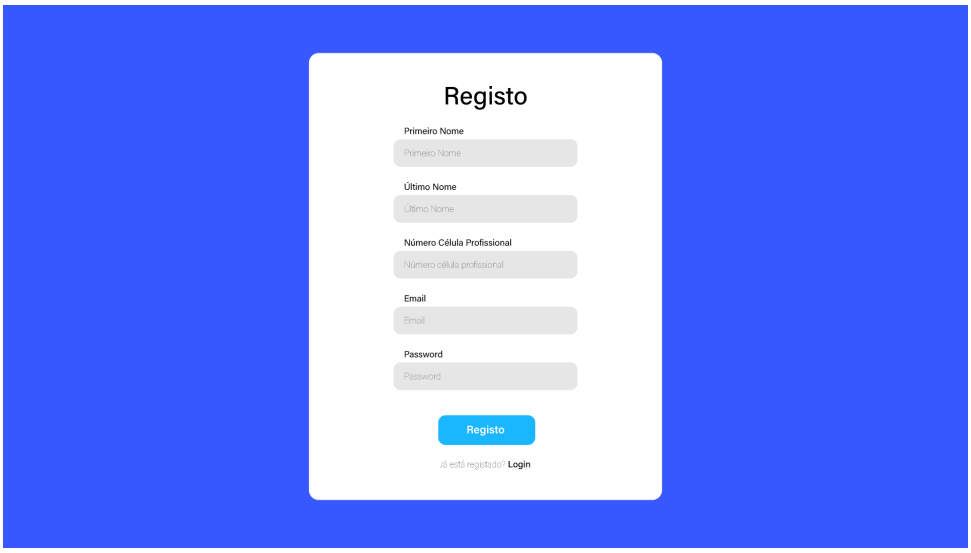


Figura 5.2: Mockup: Registo

Para efectuar o registo, o visitante necessita de preencher o seu nome e número da célula profissional, para que possa ser validado por um administrador. Precisa também de um email e password para acesso à plataforma no caso do seu registo ser aceite.

5.2.2 Utilizadores autenticados

5.2.2.1 Importar boletins

Como utilizador autenticado, este deve ser capaz de submeter novos boletim, para que os dados destes sejam extraídos.

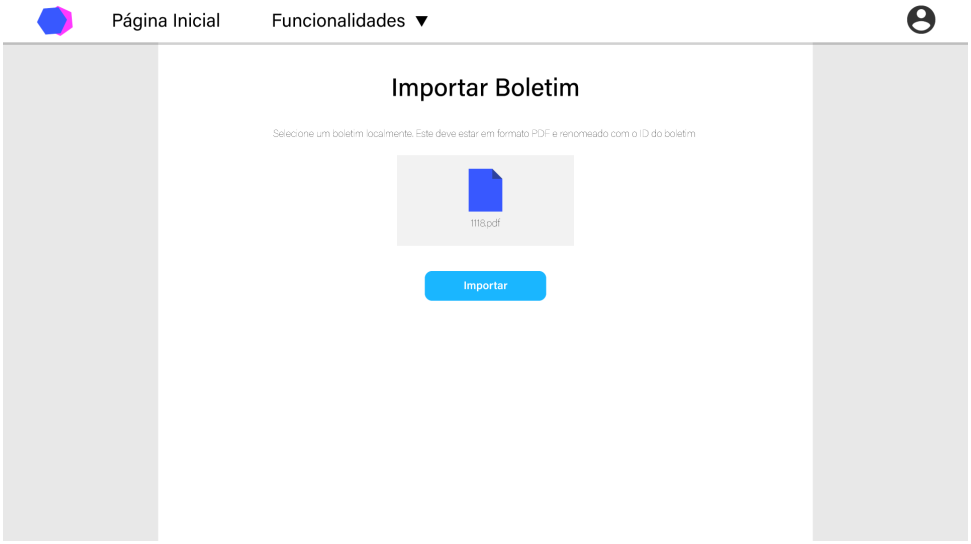


Figura 5.3: Mockup: Importar boletins

5.2.2.2 Verificar estado de um boletim

Como utilizador autenticado, o utilizador deve ser capaz de acompanhar o estado de importação de um boletim.



Figura 5.4: Mockup: Estado boletim

O utilizador deve ser capaz de conhecer a etapa atual do boletim e, em caso de conclusão da exportação, ser capaz de validar os mesmos.

5.2.2.3 Validar dados

Como utilizador autenticado, o usuário deve ser capaz de validar os dados extraídos por um boletim.

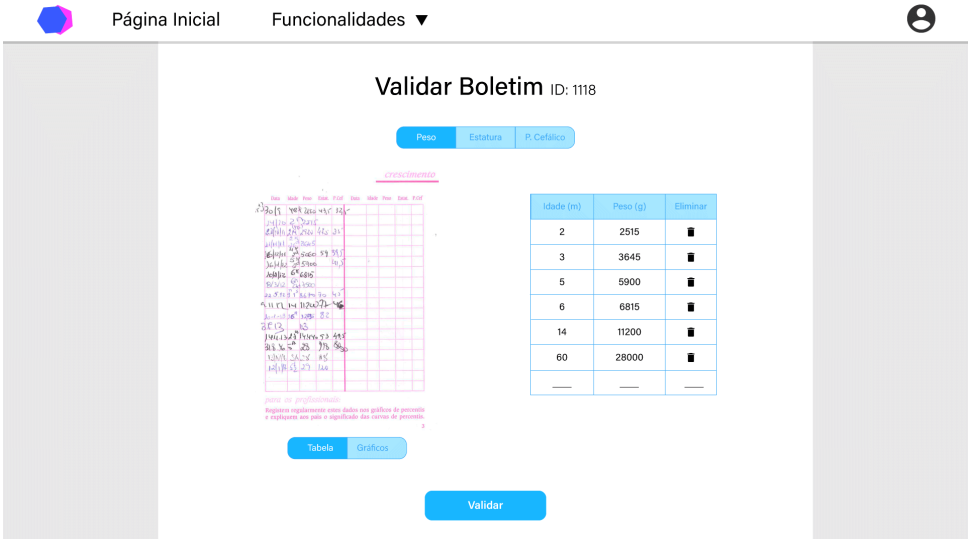


Figura 5.5: Mockup: Validar dados

O utilizador deve ter acesso aos dados extraídos de um boletim e às páginas reais deste mesmo boletim para poder fazer uma comparação e validação dos dados obtidos. Deve ser possível adicionar, remover e editar dados.

Para facilitar a validação, os dados devem ser separados por categoria - peso, estatura e perímetro cefálico - e apresentadas apenas as páginas reais do boletim referentes a cada uma delas.

5.2.2.4 Analisar dados

Como utilizador autenticado, o usuário deve ser capaz de obter e analisar os dados validados, para que possa fazer estudos e comparações com os mesmos.

O utilizador deve ser presenteado com a possibilidade de escolher e filtrar os dados que desejar e, em acréscimo, ser capaz de fazer download dos dados para poderem ser usados e analisados por outra plataforma a seu critério.

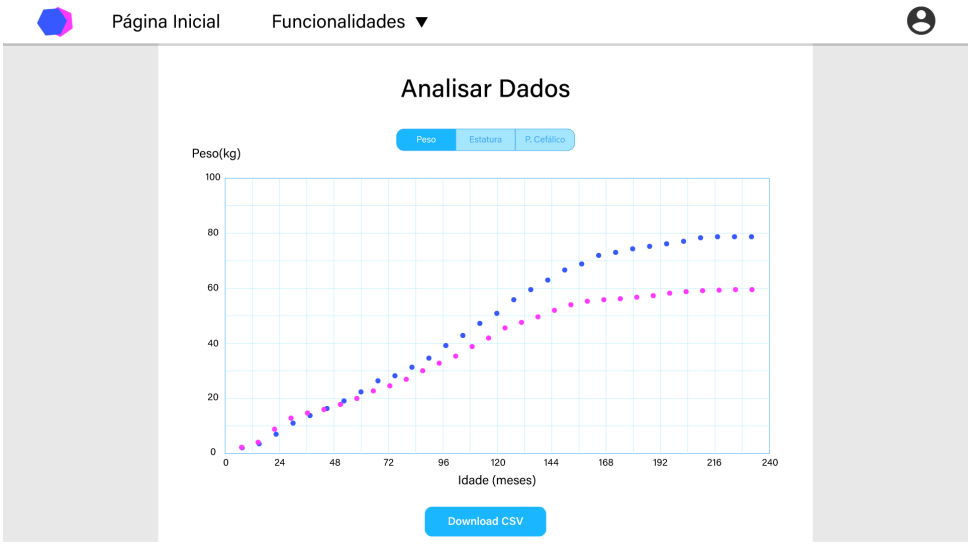


Figura 5.6: Mockup: Analisar dados

5.2.2.5 Treinar classificadores

Como utilizador autenticado, este deve ser capaz de classificar os caracteres detectados na extracção dos dados dos boletins, de modo a otimizar classificações futuras.

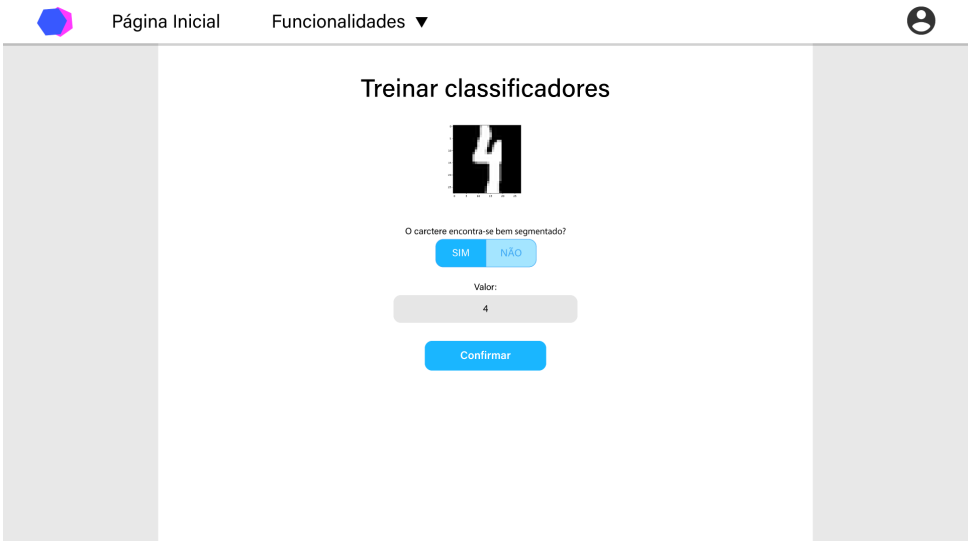


Figura 5.7: Mockup: Treinar classificadores

O utilizador deve ser capaz de, para cada segmentação, dar input sobre a boa ou má segmentação e, em caso da primeira, atribuir o valor correcto ao caractere.

5.2.3 Administradores

5.2.3.1 Validar registos

Como administrador, o utilizador deve ser capaz de analisar os pedidos de registo e aceitar ou rejeitar os mesmos, com base na sua validade.

5.2.4 Ver histórico de alterações

Como administrador, o utilizador deve ser capaz de visualizar o histórico de alteração feitas na plataforma para detectar eventuais irregularidades cometidas pelos utilizadores.

5.2.5 Eliminar utilizadores

Como administrador, o utilizador deve ser capaz eliminar um utilizador que tenha cometido irregularidades na plataforma, pondo em risco a validade dos dados.

5.3 Tecnologias

Tendo em vista a facilidade de implementação e de escalonamentos futuros, as tecnologias web são uma óptima solução para o desenvolvimento da plataforma.

Deste modo, e atendendo a que o script de exportação dos dados foi desenvolvido em Python, pode ser utilizada a *framework* Django para implementação da plataforma web. Para além de fácil utilização, permite ainda uma melhor integração com o script anteriormente desenvolvido.

Capítulo 6

Conclusões e Trabalho Futuro

Os trabalhos de investigação na área da pediatria têm por base uma enorme quantidade de dados e os Boletins de Saúde Infantil e Juvenil são uma fonte preciosa destes. A extracção destes dados levaria a uma maior facilidade de acesso a estes e, consequentemente, impulsionaria os trabalhos de investigação na área.

Com esta finalidade, esta dissertação foi realizada com o objetivo de converter automaticamente os dados manuscritos em dados digitais de fácil acesso.

Nas secções seguintes é feita uma retrospectiva do que foi feito e do que pode ser melhorado.

6.1 Trabalho Realizado

Dos objetivos inicialmente traçados, faltou somente a total implementação da interface gráfica para a análise dos resultados. Estes estão, porém, a ser estruturados e exportados para uma base de dados, facilitando o seu acesso.

Partindo da base teórica do trabalho desenvolvido anteriormente por [Ama18], foram desenvolvidos novos algoritmos para a extracção de pontos nos diferentes gráficos, não limitando aos gráficos dos boletins cor de rosa.

Quanto aos processamento das tabelas, foram desenvolvidos novos métodos para otimizar o isolamento das células individuais, mesmo em gráficos super preenchidos em que as linhas são sobrepostas por múltiplos caracteres.

Numa fase final, foi desenvolvido um algoritmo para cruzar os dados obtidos das diferentes fontes - gráficos e tabelas. Devido à falta de tempo, este algoritmo ainda se encontra numa fase inicial e precisa de futuros ajustes.

6.2 Trabalho Futuro

Em ordem de otimizar e melhorar o trabalho desenvolvido são necessárias algumas alterações ao trabalho realizado.

Um dos principais tópicos a necessitar de melhorias é a segmentação e classificação de caracteres das células da tabela. É fulcral a adição de novos classificadores de dígitos manuscritos e cruzar os resultados de cada um deles para melhorar o reconhecimento de caracteres. Quanto à segmentação, é necessária a criação de um algoritmo para lidar com os casos de má segmentação, quando os dígitos estão demasiado juntos para os isolar.

Para além disto, é necessário recolher uma maior quantidade de *templates* de pontos para otimizar a deteção de diferentes formas de pontos.

O ponto final, seria a finalização do desenvolvimento da interface gráfica, de modo a tornar todo o processo de extracção, validação e análise mais fácil e intuitivo.

Referências

[AG14] S. S. Ali e M. U. Ghani. Handwritten digit recognition using dct and hmms. In *2014 12th International Conference on Frontiers of Information Technology*, pages 303–306, Dec 2014. doi:10.1109/FIT.2014.63.

[Ama18] Fábio Amarante. *Extracting medical information from personal child health records*. PhD thesis, Faculdade de Engenharia da Universidade do Porto, Porto, 2018.

[AYV01] N. Arica e F. T. Yarman-Vural. An overview of character recognition focused on off-line handwriting, 2001. doi:10.1109/5326.941845.

[Bha14] Muzamil Bhat. Digital Image Processing. *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH*, 3(1), 2014. URL: www.ijstr.org.

[CMBK17] Arindam Chaudhuri, Krupa Mandaviya, Pratixa Badelia e Soumya K Ghosh. *Optical Character Recognition Systems for Different Languages with Soft Computing*. 2017. doi:10.1007/978-3-319-50252-6.

[dS] Direção Geral da Saúde. Boletim de saúde infantil e juvenil. Disponível em https://www.dgs.pt/paginas-de-sistema/saude-de-a-a-z/boletim-de-saude-infantil.aspx.

[FSF11] Cristina Freitas, Helena Sofia Sousa e Helena Fonseca. Boletim de Saúde Infantil e Juvenil: o exame global de saúde dos 11 aos 13 anos (parte II). *Revista Portuguesa de Clínica Geral*, 27:193 – 202, 03 2011. URL: http://www.scielo.mec.pt/scielo.php?script=sci_arttext&pid=S0870-71032011000200008&nrm=iso.

[GSGN17] Angelos P. Giotis, Giorgos Sfikas, Basilis Gatos e Christophoros Nikou. A survey of document image word spotting techniques. *Pattern Recognition*, 68:310 – 332, 2017. URL: http://www.sciencedirect.com/science/article/pii/S0031320317300870, doi:https://doi.org/10.1016/j.patcog.2017.02.023.

[GW07] Rafael C Gonzalez e Richard E Woods. *Digital Image Processing*. 2007.

[HKP11] Jiawei Han, Micheline Kamber e Jian Pei. Data Mining. Concepts and Techniques, 3rd Edition (The Morgan Kaufmann Series in Data Management Systems). Technical report, 2011.

[JP05] João José e Rodrigues Pereira. Universidade do Minho Escola de Engenharia Modelos de Data Mining para multi-previsão: aplicação à medicina intensiva. Technical report, 2005.

[MAN+14] Gupta Mehul, Patel Ankita, Dave Namrata, Goradia Rahul e Saurin Sheth. Text-based Image Segmentation Methodology. *Procedia Technology*, 14:465–472, 2014. doi:10.1016/j.protcy.2014.08.059.

[MNY99] Shunji Mori, Hirobumi Nishida e Hiromitsu Yamada. *Optical Character Recognition*. John Wiley & Sons, Inc., USA, 1st edition, 1999.

[Nun06] Fátima L S Nunes. Introdução ao Processamento de Imagens Médicas para Auxílio ao Diagnóstico – Uma Visão Prática. *Atualizações em Informática*, 2006.

[RBD+17] Partha Roy, Ayan Bhunia, Ayan Das, Prithviraj Dhar e Umapada Pal. Keyword spotting in doctor's handwriting on medical prescriptions. *Expert Systems with Applications*, 76:113–128, 06 2017. doi:10.1016/j.eswa.2017.01.027.

[RLSG16] George Retsinas, Georgios Louloudis, Nikolaos Stamatopoulos e Basilis Gatos. Keyword Spotting in Handwritten Documents Using Projections of Oriented Gradients. In *Proceedings - 12th IAPR International Workshop on Document Analysis Systems, DAS 2016*, 2016. doi:10.1109/DAS.2016.61.

[SF16] Sebastian Sudholt e Gernot Fink. Phocnet: A deep convolutional neural network for word spotting in handwritten documents. pages 277–282, 10 2016. doi:10.1109/ICFHR.2016.0060.

[SMS12] Nikita Sharma, Mahendra Mishra e Manish Shrivastava. COLOUR IMAGE SEGMENTATION TECHNIQUES AND ISSUES: AN APPROACH. *International Journal of Scientific & Technology Research*, 1(4), 2012. URL: www.ijstr.org.