



João Miguel Carvalho Nunes

Licenciado em Ciências da Engenharia Electrotécnica e
Computadores

Leitura Automática de Documentos para Sistemas de Informação

Dissertação para obtenção do Grau de Mestre em
Engenharia Electrotécnica e de Computadores

Orientador: João Rosas, Professor Auxiliar, FCT-UNL

Co-orientador: Eng. Tiago Duarte, Softconcept

Júri:

Presidente: Doutor Luís Filipe Figueira de Brito Palma - FCT/UNL

Arguente(s): Doutor Miguel Jorge Tavares Pessoa Monteiro - FCT/UNL

Vogal(ais): Doutor João Almeida das Rosas - FCT/UNL

Engenheiro Tiago Macara Duarte - Empresa SoftConcept

FCT FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Setembro 2013

Copyright ©

Leitura automática de documentos para Sistemas de Informação

João Miguel Carvalho Nunes - FCT/UNL - UNL

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Agradecimentos

Tenho a agradecer, em primeiro lugar, ao Professor João Rosas pela grande disponibilidade e apoio dados ao longo deste percurso.

Agradeço o apoio logístico fornecido pela Softconcept, e a inspiração e ideias motivadoras do seu representante Eng. Tiago Duarte.

Devo também um agradecimento ao Departamento de Engenharia Eletrotécnica da FCT/UNL pelas muitas horas, por vezes tardias, que me possibilitou trabalhar.

À família e amigos por todo o apoio, motivação e paciência para me suportar tantas vezes a falar de um ou vários temas tão distintos dos seus interesses. Um agradecimento especial aos meus pais, sem os quais todo este percurso não seria possível.

Aos amigos que me ajudaram a chegar até este patamar final e me auxiliaram na realização da tese, tais como, amigos de curso como o Rui Medeiros e Tiago Gonçalves. E também a todos os outros que me ajudaram tanto com a sua motivação, como também com momentos lúdicos que me permitiram voltar sempre com mais força ao trabalho.

E por fim, um agradecimento especial à minha namorada por toda a paciência, carinho e motivação para ultrapassar esta etapa.

Resumo

Nos dias que correm, é cada vez mais uma necessidade o acesso à informação de forma rápida e eficaz. Pelo que, para simplificar esse acesso, toda a documentação é digitalizada, mas muita da informação original continua a existir em formato físico. Um exemplo dessa situação são os documentos pessoais de identificação, ou seja, são documentos digitalizados mas continuam a ser necessários encontrar-se em formato físico.

O problema coloca-se quando existe a necessidade de ler esses documentos, dado que actualmente não há forma automática de o fazer, ou pelo menos, que seja suficientemente eficaz face ao processo manual.

Contudo, este processo apesar de garantir a fiabilidade na compreensão e transição dos dados, torna-se muito pouco eficiente. É um grande consumidor de recursos humanos, devido à sua morosidade e carácter repetitivo. Um caso concreto deste problema, é a acção de “check-in no hotel”, onde existe a necessidade da introdução rápida, exacta e a mais automatizada possível de toda a informação respeitante aos seus clientes. O processo actual não é rápido, nem tão pouco eficiente, pois consiste na introdução manual dos dados visualizados pelos operadores.

Este trabalho contribui para encontrar uma solução em que se recorra somente à digitalização de um documento, ao reconhecimento óptico do texto e sua interpretação, de forma a simplificar o processo. Automatizando o mesmo por completo, de forma a ser somente executado por uma máquina, necessitando apenas de uma intervenção mínima por parte do operador.

Para atingir esse objectivo, procedeu-se em primeiro lugar a uma investigação sobre as diversas tecnologias, de processamento de imagem, reconhecimento de padrões, recolha de informação e tratamento de dados, existentes actualmente. Com vista à selecção das melhores abordagens para posterior integração, de forma a alcançar-se o objectivo pretendido para este projecto.

Deste trabalho resulta um “software”, que realiza a identificação e leitura automática de documentos, recolhendo a sua informação útil de forma estruturada.

Palavras-Chave: Identificação automática de documentos, reconhecimento de texto e caracteres (OCR), Reconhecimento de Padrões (RP), Leitura Automática de Documentos (LAD).

Abstract

Nowadays there is an increasing necessity to access information in a quick and effective way. Therefore, to make access easier a lot of documentation is nowadays scanned, but always assuring the original information still exists in physical format. Take for instance personal identification documents, they can be scanned but they are required to exist as a physical object.

Although it is easy to convert documents to a digital format, reading their content and using it is not. Currently there is no automatic way to do this, or there is no way which is as effective as manual insertion.

Manual insertion of data assures the data is correctly understood and inserted but is not very efficient. It is a very repetitive, time consuming process and therefore a waste of human resources.

A case of this problem is the action of 'checking-in at an hotel', where there is the need to introduce all information regarding the customers as fast, accurate and automated as possible. The current process is not quick, nor efficient, as it relies on manual entry of data read by the operators.

This work aims at producing a solution to simplify the process that consists only of scanning a document to retrieve the information contained in it. For that purpose the process was completely automated, requiring only a scanning equipment and minimal intervention from the operator.

To achieve this goal the different technologies that were to be integrated: image processing, pattern recognition, data collection and data processing were firstly identified and evaluated. Then the best options were selected for further integration, so as to reach the desired goal for this project.

As an output of this work software that can identify and read documents automatically, collecting the useful information in a structured way, was created.

Key-Words: Automatic document identification, text and characters recognition, automatic document reader, pattern recognition.

Índice

1. Introdução.....	1
1.1. Motivação	1
1.2. Objectivos.....	3
1.3. Contribuições originais	4
1.4. Estrutura da dissertação.....	5
2. Revisão literária	7
2.1. Técnicas de representação de imagem	7
2.1.1. Imagem digital	7
2.1.2. Binarização.....	8
2.1.3. Sistema RGB.....	8
2.1.4. Escala de cinzentos (Grey scale).....	9
2.2. Técnicas de análise e manipulação de imagem.....	10
2.2.1. Histograma	10
2.2.2. Haar Wavelet Transform.....	10
2.2.3. Haar Cascade Classifier	11
2.2.4. Componentes Ligados.....	14
2.3. Sistemas “OCR”	15
2.4. Leitura automática de documentos	17
2.5. Reconhecimento de padrões	18
2.5.1. Rough Sets.....	19
2.5.2. Redes Neuronais.....	21
2.5.3. Redes ou Mapas de Kohonen	24
3. Desenvolvimento do sistema LAD	27
3.1. Modelação do sistema.....	27
3.1.1. Ilustração do Conceito de LAD	27
3.1.2. Requisitos Funcionais.....	31
3.1.3. Requisitos Não Funcionais	34

3.2. Especificação do Sistema de Leitura Automática de Documentos	34
3.3. Implementação do sistema LAD	37
3.3.1. Arquitetura do sistema LAD	37
3.3.2. Leitura (OCR) e interpretação de documentos	41
3.3.3. Mecanismo de identificação dos modelos baseado em Rough Sets	46
3.4. Arquitetura de Software	49
3.4.1. Aquisição de imagem e pré-tratamento	50
3.4.2. Equipamento utilizado no desenvolvimento	54
3.5. Sistema LAD desenvolvido	55
3.6. Testes e Validação	58
3.6.1. Verificação	58
3.6.2. Validação	61
3.7. Resultados experimentais	63
4. Conclusões	65
4.1. Síntese do trabalho efectuado	65
4.2. Trabalho Futuro	67
Bibliografia	69
Anexos	73

Índice de Figuras

Figura 1.1 - Exemplo de diversidade de documentos de identificação	3
Figura 2.1 - Exemplo de imagem digital	7
Figura 2.2 - Imagem 2.1 binarizada	8
Figura 2.3 - Sistema de cores Red-Green-Blue (R-G-B)	9
Figura 2.4 - Imagem 2.1 em escala de cinzento.....	9
Figura 2.5 - Exemplo de histograma	10
Figura 2.6 - Haar-Like features	11
Figura 2.7 - Processamento de imagem aplicando Haar-like features	12
Figura 2.8 - Algoritmo de identificação de faces em funcionamento.....	13
Figura 2.9 - Resultado final do processamento da imagem recorrendo a Haar-like features.....	13
Figura 2.10 - Mascara para aplicação do algoritmo CC com vizinhança a 4 pixels	14
Figura 2.11 - Mascara para aplicação do algoritmo CC com Vizinhança a 8 pixels.....	15
Figura 2.12 - Modelo de funcionamento de um OCR	16
Figura 2.13- Estrutura de funcionamento do Tesseract (Smith et al. 2009)	16
Figura 2.14 - Modelo genérico para definir a estrutura dum documento “(Kauniskangas 1999)”.....	17
Figura 2.15 - Rough Set (Pawlak & Skowron 2007)	20
Figura 2.16 - Exemplo de Rede Neuronal no cérebro humano	21
Figura 2.17 - Estrutura de uma Rede Neuronal (Chakraborty 2010)	22
Figura 2.18 - Gráfico tipologias de Redes Neuronais	23
Figura 2.19 - Exemplo de Kohonen Neural Network	25
Figura 3.1 - Exemplo de inserção de dados do proprietário de um veículo por um utilizador humano.	28
Figura 3.2 - Obtenção de informação através de LAD	28
Figura 3.3 - Exemplo de documento e campos a recolher (Rankl & Effing 2010).....	29
Figura 3.4 - Cartão do cidadão (Ministros, 2009)	29
Figura 3.5 - Processo de Leitura Automática de Documentos.....	30
Figura 3.6 - Exemplo de recolha de informação de um documento de identificação ..	30
Figura 3.7 - Processo de leitura dum documento em suporte físico	32
Figura 3.8 - Identificação manual dum documento.....	32
Figura 3.9 - Identificação automática dum documento	32
Figura 3.10 - Processo de leitura interpretada dum documento	33
Figura 3.11 - Processo de criação manual dum novo modelo.....	33

Figura 3.12 – Processo de criação automática de um novo modelo	33
Figura 3.13 - Definição do modelo de um documento	35
Figura 3.14 - Arquitetura global do sistema.....	37
Figura 3.15 - Arquitetura do sistema quando não é identificado documento	38
Figura 3.16- Processo mais comum de digitalização de imagem	39
Figura 3.17 – Processos que definem o bloco Scan	39
Figura 3.18 – Processos que definem o bloco Avaliador.....	40
Figura 3.19 - Modelo Inicial Sistema LAD	41
Figura 3.20 - Modelo Final Sistema LAD.....	42
Figura 3.21 - Processo de recolha de características do documento	43
Figura 3.22 - Processamento de imagem aplicando Haar-like features	43
Figura 3.23 - Diagrama sequência UML do processo de funcionamento do software .	45
Figura 3.24 - Visão global da arquitetura e bibliotecas usadas para a implementar	50
Figura 3.25 - Funções OpenCV usadas	51
Figura 3.26 - Funções Aforge usadas	51
Figura 3.27 - Funções desenvolvidas com apoio da ferramenta Rosetta	52
Figura 3.28 - Funcionalidades usadas recorrendo a Prolog	53
Figura 3.29 - Plataformas tecnológicas usadas.....	53
Figura 3.30 - Ambiente de desenvolvimento montado para a implementação do software.....	54
Figura 3.31 - Scanner utilizado no projecto	54
Figura 3.32 - Layout global da ferramenta	55
Figura 3.33 - Menus de funcionalidades do software	56
Figura 3.34 - Exemplo de aplicação de CC recorrendo à ferramenta desenvolvida	56
Figura 3.35 - Exemplo de leitura executada sobre um documento de identificação com o software criado durante este projecto	57
Figura 3.36 - Arquitetura em que cada componente pode ser substituída.....	57
Figura 3.37 - Documento em suporte físico.....	58
Figura 3.38 - Documento em suporte digital	59
Figura 3.39 - Identificação do tipo de documento.....	59
Figura 3.40 - Exemplo de modelo de documento.....	60
Figura 3.41 - Gáfico de Avaliação de desempenho do software a identificar documentos	62
Figura 3.42 - Gráfico de desempenho do bloco OCR conforme a biblioteca aplicada .	62

Índice de Tabelas

Tabela 1.1 - Estrutura da Dissertação.....	5
Tabela 2.1 - Exemplo de dados usados numa rede neuronal de teste	24
Tabela 3.1- Requisitos funcionais para o sistema LAD	31
Tabela 3.2 - Requisitos Não Funcionais.....	34
Tabela 3.3 - Conjunto de treino.....	46
Tabela 3.4 - Tabela de decisão.....	47
Tabela 3.5 - Classes equivalentes	47
Tabela 3.6 - Matriz de discirnibilidade	48
Tabela 3.7 - Lista parcial das regras RS e as estatísticas associadas	49
Tabela 0.1 - Tabela de regras usada no software para classificação dos documentos	75

Acrónimos

BMP - Bitmap

BOW - Bag Of Words

BP - Back-Propagation

CC - Connected Components

ESA - Explicit Semantic Analysis

GIF - Graphics Interchange Format

GUI - Graphical User Interface

IMG - Imagem Digital

LAD - Leitura Automática de Documentos

LSA - Latent Semantic Analysis

OCR - Optical Character Recognition

RGB - Red Green Blue

RN - Redes Neuronais

RP - Reconhecimento de Padrões

RS - Rough Sets

SI - Sistema de Informação

UML - Unified Modeling Language

1. Introdução

1.1. Motivação

Atualmente, em todas as atividades humanas, existe a necessidade de arquivar e gerir grandes quantidades de informação. Toda esta informação pode estar definida em diversas formas, como por exemplo: escrita, oral ou gráfica que por sua vez podem estar em suportes ditos físicos ou eletrónicos.

A tendência corrente consiste na utilização de Sistemas de Informação (SI) para guardar essa informação, pois esta forma é de mais fácil acesso, gestão e utilização. No entanto, existem ainda muitos repositórios em suporte físico. Note-se também que muitos destes documentos têm de continuar simultaneamente a existir no seu formato físico e digital, por motivos legais ou de segurança. Muitas atividades humanas, por exemplo as comerciais, continuam a utilizar o suporte em papel, não tendo ainda adotado tecnologias de informação.

Este contexto de dualismo entre informação em suporte físico e digital dá origem, a que muitas vezes, seja necessário proceder à transferência de informação do suporte físico para digital. Mas, devido ao volume dessa informação, essa transferência só poderá ser feita através de ferramentas que procedam a uma conversão física/digital de forma automática, que de outra forma seria impraticável. Neste contexto, não se trata apenas da simples leitura do documento para documento de imagem (ex: GIF ou BMP). Para que a informação possa ser facilmente acessível, é necessário sujeitar os documentos a um reconhecimento ótico dos caracteres (OCR), para assim se poder extrair o texto ou a informação neles contidas. Esta técnica designa-se por “Optical Character Recognition” (OCR). (Mori et al. 1992).

Hoje em dia, já é possível utilizar ferramentas OCR que estão disponíveis na internet ou podem ser fornecidas na aquisição de equipamentos “scanner”. No entanto, as funcionalidades dessas ferramentas resumem-se quase exclusivamente à conversão de imagem para texto. No entanto, em muitas situações é necessário ir mais além da mera obtenção de informação, e identificar qual a informação que está a ser convertida, por exemplo, extrair o conteúdo de blocos de texto num bilhete de identificação, ou num documento de venda (fatura). Ou seja, é necessário obter a informação, o seu significado e respectiva estrutura.

Esta necessidade pode ser ilustrada pelo seguinte exemplo, suponhamos que numa empresa se pretende digitalizar os documentos de venda. Utilizando as respectivas ferramentas OCR, obter-se-ia o conteúdo dos documentos, mas sem a identificação da informação obtida, por exemplo, o nome do cliente, endereço e itens vendidos. Neste sentido, a criação de um modelo e respectiva ferramenta com a capacidade de fornecer o conteúdo dos documentos, duma forma estruturada e interpretada, teria um impacto determinante na digitalização da informação e na sua utilidade. Idealmente, tal ferramenta deverá ser flexível para uma grande variedade de documentos, podendo processar novos documentos. Portanto, trata-se da designada leitura automática de documentos (LAD).

Este projecto pretende preencher a lacuna tecnológica acima identificada, relativamente há inexistência de ferramentas que implementem “OCR” e que permitam extrair a informação dos documentos, bem como o significado da informação obtida. Pretende-se assim ir além do que as tecnologias atuais permitem, a conversão de informação presente no documento para formato digital, e permitir a identificação semântica de contida nessa informação.

Neste sentido, foi desenvolvido um esforço de integração de diversas tecnologias, por forma a construir-se um sistema que:

- Digitalize documentos.
- Os consiga identificar e obter o respectivo texto do documento duma forma estruturada.
- Obtenha também a semântica da informação.

Conseguir-se-ia assim, por exemplo numa factura, identificar o cliente e os respectivos itens vendidos; num passaporte, seria possível obter os dados de identificação de uma pessoa, conforme ilustrado na Figura 1.1.



Figura 1.1 - Exemplo de diversidade de documentos de identificação

Nesta colaboração com a Softconcept pretendeu-se que a ferramenta a criar, pudesse ser usada em hotelaria para auxiliar no processo de *check-in* de clientes, sendo este o objetivo pretendido para a ferramenta criada em colaboração com a Softconcept. (Softconcept n.d.).

O avanço das tecnologias de reconhecimento de caracteres, bem como da Inteligência Artificial, permitem já uma grande eficiência no tratamento de documentos e conversão para formato digital. No entanto, ainda existe bastante trabalho a efetuar em termos da semântica do conteúdo obtido em resultado de uma digitalização.

Este aspeto torna este projeto mais aliciante, pois possui uma significativa componente prática, permitindo construir e validar o modelo inerente à ferramenta e certificar a utilidade das conversões efetuadas..

1.2. Objectivos

O objectivo principal desta dissertação de mestrado é a obtenção de um processo de leitura automática de documentos. Esta leitura tem em vista fornecer informação estruturada, desses documentos, para utilização em SI.

Para alcançar este objectivo principal, foram estabelecidos objectivos intermédios com vista a conseguirmos uma progressão mais segura:

- Identificação e investigação das tecnologias mais adequadas para suportar e concluir, com êxito, a tarefa principal.
- Escolha e decisão pelas tecnologias mais adequadas, a integrar de forma a permitir a identificação automática dos diferentes tipos de documentos e respectivas informações
- Definição da estrutura dos modelos de tipo de documento que permita fornecer uma informação estruturada aos SI

1.3. Contribuições originais

A tese apresenta contribuições relevantes no desenvolvimento científico, não tanto pelo seu cariz inovador ou criação de novas tecnologias, mas sim pela forma como foram utilizadas e pelos objectivos alcançados com as mesmas.

A principal contribuição deste projecto consiste na criação de um módulo de leitura automática de documentos que permite obter uma arquitectura modelar.

A nível científico a grande contribuição surge da utilização do algoritmo de Haar Cascade classifier (classificador de Haar em cascata) para identificação de faces, provando a sua robustez e eficácia pelos resultados obtidos.

O elemento mais inovador e a maior contribuição científica, advém da utilização de Rough Sets para classificação de documentos, demonstrando todo o seu potencial com a grande precisão dos resultados obtidos. Comprovou-se ainda a sua versatilidade ao permitir a aprendizagem de novas tipologias de documentos de forma simples e clara.

Outro contributo importante, que também ajudou ao sucesso do projecto, foi a selecção e teste do conjunto mais acertado de características, que permite a identificação de documentos pessoas.

.

1.4. Estrutura da dissertação

Este trabalho começa por fazer uma revisão do estado-da-arte relativo às técnicas de aquisição de imagem, métodos e sistemas OCR, leitura automática de documentos e métodos de reconhecimento de padrões para assim podermos contextualizar este projecto dentro do âmbito das tecnologias da informação.

Na fase seguinte descreve-se a especificação da ferramenta de leitura automática de documentos proposta neste trabalho, que é seguida pela descrição do seu correspondente desenvolvimento. Seguidamente procede-se à validação e verificação da ferramenta desenvolvida, sendo que após esta fase, faz-se uma correspondente análise de resultados. Finalmente, e após uma síntese do trabalho e resultados obtidos, descrevem-se alguns projetos a desenvolver em trabalho futuro. A tabela 1.1. descreve a estrutura e os tópicos abordados em cada capítulo desta dissertação.

Tabela 1.1 - Estrutura da Dissertação

Capítulo	Tema	Resumo
1	Introdução	Breve resumo sobre o trabalho desenvolvido, motivação e contextualização, explicação da estrutura do documento
2	Estado da Arte	Resumo de tecnologias já existentes que se aproximem ou adequem ao tema em estudo/análise. São explicados todos os conceitos necessários para a compreensão do trabalho
3	Desenvolvimento do Sistema LAD	Explicitado o percurso feito ao longo do trabalho, os desenvolvimentos alcançados, as opções tomadas e o que realmente se obteve. Demonstração dos resultados perante as opções tomadas e explicadas no capítulo anterior
4	Conclusão	Síntese do trabalho efectuado. Analise dos resultados obtidos e trabalho futuro

2. Revisão literária

Nesta secção faz-se uma apresentação do estado-da-arte dos aspectos importantes para o trabalho desenvolvido. Mais concretamente são explicadas as tecnologias OCR e a leitura automática de documentos, começando com alguns conceitos e técnicas de representação e tratamento de imagem digital.

2.1. Técnicas de representação de imagem

2.1.1. Imagem digital

Uma imagem digital (IMG) (Figura 2.1) mais não é que um mapa bidimensional de uma imagem captada por um sensor, onde cada coordenada do mapa corresponde a um pixel com um valor RGB, que permite saber que cor foi recolhida pelo sensor naquele ponto, para essa IMG (Pratt 2007; Hirayama et al. 2011; Gonzalez & Woods 2002).



Figura 2.1 - Exemplo de imagem digital

2.1.2. Binarização

A binarização consiste num processo de conversão de uma imagem colorida para preto e branco, recorrendo a algoritmos que estabelecem um valor limite de cor para discriminar quais os bites que ficam a preto ou a branco (Gonzalez & Woods 2002), conforme ilustrado na Figura 2.2. O valor que define este limite é denominado de limiar. Existem vários métodos e algoritmos para definir este valor, recorrendo todos à análise dos níveis de cores ao longo da imagem, embora alguns o façam diferenciando zonas, dando assim origem a um limiar dinâmico, enquanto outras analisam a imagem como um todo.

O método de binarização mais utilizado é o método de Otsu (Otsu 1975) que permite identificar o valor potencialmente ideal para estabelecer este limiar.



Figura 2.2 - Imagem 2.1 binarizada

2.1.3. Sistema RGB

O sistema de cores utilizado no tratamento de IMG chama-se RGB. Esta sigla corresponde às iniciais de Red (vermelho), Green (verde), e Blue (azul). Estas são as cores básicas para a formação de todas as outras (Figura 2.3), ou seja, todas as outras cores podem ser conseguidas combinando uma determinada quantidade de

vermelho, verde e azul, sendo essa a quantidade que é definida nos pixéis de uma IMG digital.

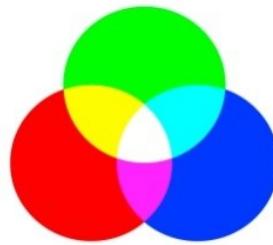


Figura 2.3 - Sistema de cores Red-Green-Blue (R-G-B)

2.1.4. Escala de cinzentos (Grey scale)

A escala de cinzentos é uma representação da imagem recorrendo somente a tons cinzentos. Trata-se de um regime de cores que permite acentuar contrastes, muito útil para reconhecimento de texturas.

O tom de cada pixel é obtido pela média dos valores das componentes RGB, $\frac{R+G+B}{3}$, após a conversão de todos os pixéis conseguimos então uma imagem em escala cinzentos, conforme ilustrado na Figura 2.4 (Gonzalez & Woods 2002).



Figura 2.4 - Imagem 2.1 em escala de cinzento

2.2. Técnicas de análise e manipulação de imagem

2.2.1. Histograma

Gráfico estatístico no qual se apresenta o número de vezes que determinado nível de cor aparece em todos os pixéis de uma IMG. A escala de valores da cor pode variar entre os valores 0 e 255 (Figura 2.5). Sendo o valor 0 correspondente à cor preta e o valor 255 à cor branca, que são os extremos da escala numérica.

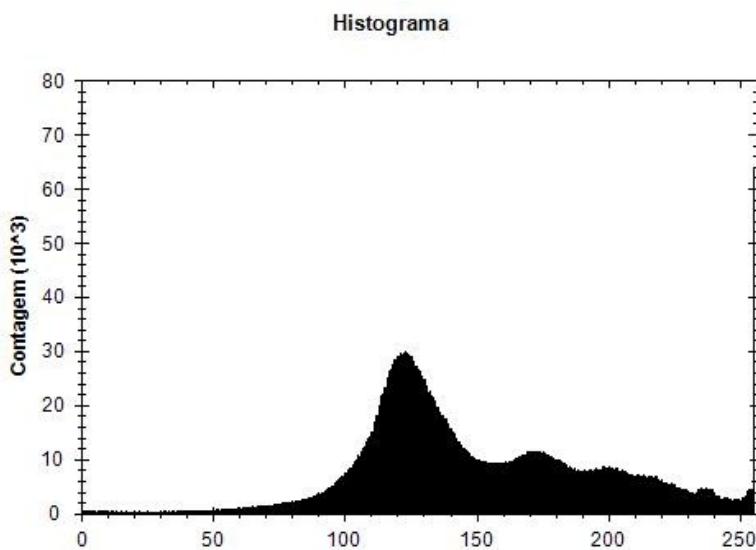


Figura 2.5 - Exemplo de histograma

2.2.2. Haar Wavelet Transform

Trata-se de um conjunto de transformadas aplicadas a formas de onda, proposto por Alfred Haar em 1909. São usadas como ferramenta matemática de decomposição de informação de forma hierárquica, podendo ser aplicadas para vários fins. A Haar wavelet é a forma de onda mais simples com a qual se pode trabalhar, que é definida pela expressão que se segue na Equação 1.

$$\psi(x) := \begin{cases} 1 & \text{for } 0 \leq x < 1/2 \\ -1 & \text{for } 1/2 \leq x < 1 \\ 0 & \text{otherwise.} \end{cases}$$

Equação 1

Trata-se de uma forma de simplificar o processo de transmissão de informação. No caso específico do tratamento de imagem, é usado para a compressão da informação. A imagem deixa de ser caracterizada pelos seus pixéis para ser definida em coeficientes de onda, pois este algoritmo permite descrever uma imagem e o seu conteúdo de uma forma mais leve e fácil de processar (Stollnitz et al. 1995; Way et al. 2004; Viola & Jones 2004; Struzik et al. 1999).

Dada a sua utilidade esta transformada é usada como base no Haar Cascade Classifier, que faz a identificação de caras em imagens, conforme descrito mais abaixo.

2.2.3. Haar Cascade Classifier

Consiste num classificador simples, que aplica a toda uma imagem as haar-like features (Figura 2.6), que são características que permitem identificar zonas específicas de uma imagem. Conforme o nome indica, trata-se de um sistema em cascata, significa que se aplica o classificador simples recursivamente, aplicando-se as features sequencialmente em sub-regiões da IMG (Figura 2.7), até o algoritmo identificar o fim de todas as etapas, ou até ser rejeitado pelo classificador em alguma das etapas.

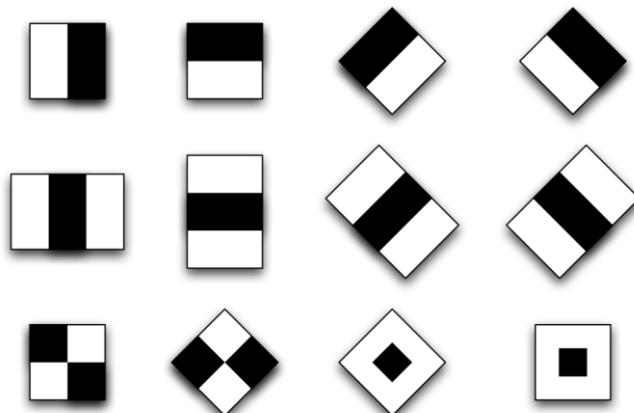


Figura 2.6 - Haar-Like features

Este classificador permite a identificação de faces humanas numa imagem a preto e branco (Figura 2.7). Sendo actualmente um dos processos mais utilizados, dada a possibilidade de ser implementado a partir da biblioteca Open CV e a sua grande eficácia. Dependendo do que se pretende encontrar e das dimensões da imagem, as características são adaptáveis, sendo aplicadas variações no tamanho e rotações na mesma para se encontrar correspondências.

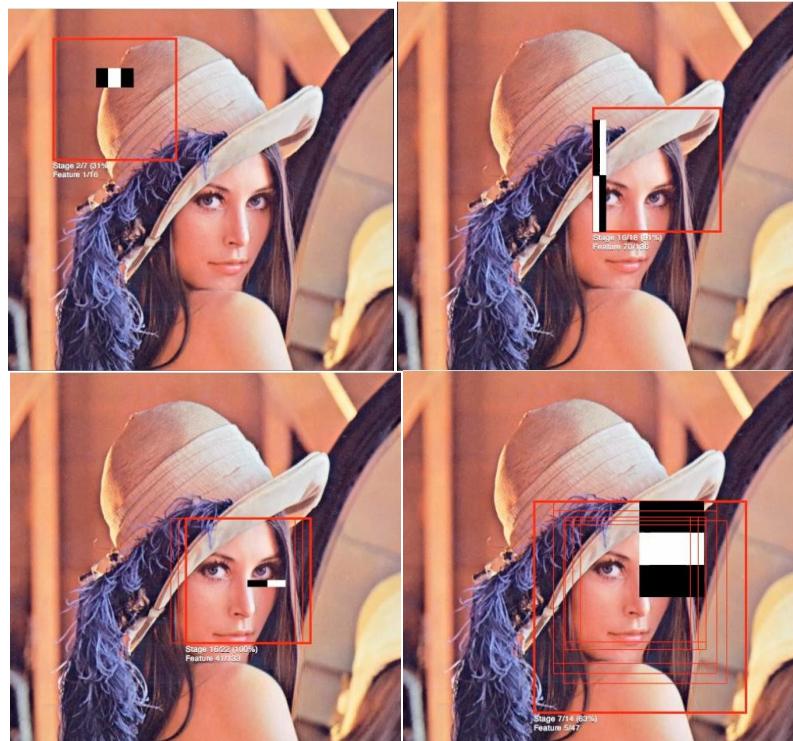


Figura 2.7 - Processamento de imagem aplicando Haar-like features

Este algoritmo trabalha mais exaustivamente nos locais onde se consegue encontrar correspondências e aí aplica variações nas features e conforme descrito anteriormente, também elas são utilizadas consoante a posição na imagem e na sub-região em que estão a ser aplicadas. Por exemplo, sabe-se que os olhos estão acima do nariz, logo a característica que potencialmente irá identificar os olhos não será a mesma que a que se aplicará para o nariz. Conforme, se pode ver no exemplo das figuras anteriores, tem-se as haar-like features específicas para determinadas zonas do rosto. (Viola & Jones 2004; Way et al. 2004; Lienhart & Maydt 2002)



Figura 2.8 - Algoritmo de identificação de faces em funcionamento

Na Figura 2.8 tem-se um exemplo do algoritmo em funcionamento, observando-se as Haar-like features a encontrar correspondências na imagem cujo resultado se pode constatar na Figura 2.9.



Figura 2.9 - Resultado final do processamento da imagem recorrendo a Haar-like features

2.2.4. Componentes Ligados

Apresenta-se agora um método de segmentação e identificação do conteúdo duma imagem.

Processo que divide a imagem em pequenos blocos de forma a facilitar o reconhecimento e definição do conteúdo presente

Os componentes ligados são um algoritmo de processamento de imagem a nível do pixel, que têm como objetivo a segmentação da imagem e identificação de blocos dentro da imagem (Gonzalez & Woods 2002). Para tal, recorre a um método que realiza a verificação repetidamente até não serem procedidas mais alterações.

O método divide-se em dois passos, o primeiro passo é a verificação se um determinado pixel é preto ou branco. Se for preto, significa que tem conteúdo, atribuindo-lhe por isso um identificador, ficando assim todo os pixéis pretos catalogados com um identificador diferente e único. O segundo passo é aplicado recursivamente e após identificação de todos os pixéis com conteúdo, que se agrupam por uma ligação, dando origem ao nome Componentes Ligados. São identificadas as ligações segundo uma vizinhança específica que pode ser a 4 ou 8 pixéis, como se pode ver na Figura 2.10 e Figura 2.11 respectivamente.

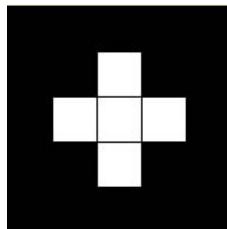


Figura 2.10 - Mascara para aplicação do algoritmo CC com vizinhança a 4 pixels

Este segundo passo consiste numa verificação em redor de cada pixel, para verificar se existe à sua volta algum pixel com uma etiqueta menor que a sua, passando essa a ser a sua etiqueta. Este processo é executado consecutivamente até que não seja necessário realizar mais nenhuma alteração de etiqueta.

Quando não se verificarem mais alterações de etiquetas, o algoritmo sabe que chegou ao fim e já foi executada a segmentação e identificados os blocos de pixéis ligados entre si.

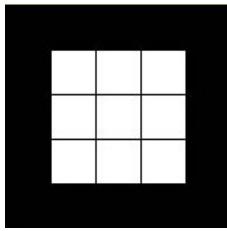


Figura 2.11 - Mascara para aplicação do algoritmo CC com Vizinhança a 8 pixels

2.3. Sistemas “OCR”

O sistema de identificação de caracteres (OCR) é uma tecnologia utilizada na conversão de texto, que se encontra dentro de imagens, para um formato digital. Trata-se de uma tecnologia já com uma significativa maturidade, dado que o seu estudo e desenvolvimento começou há décadas. Um exemplo desta tecnologia é o mecanismo Tesseract, desenvolvido nos laboratórios da HP (Smith 2007) desde 1984, já nessa altura, para competir com outros softwares existentes no mercado. Segundo (Mori et al. 1992) os primeiros avanços destas tecnologias começaram nos anos 50, inicialmente apenas conceptualmente, mas gradualmente avançando para ferramentas mais concretas. As primeiras versões eram bastante falíveis, baseando-se na mera comparação de documentos ou, mais tarde, em imagens como Modelos (Mori et al. 1992).

Com o passar do tempo concluiu-se que a comparação com Modelos não era suficiente para obter bons resultados e começou-se a investigar a possibilidade de usar análise estrutural dos caracteres para a identificação dos mesmos. A partir dessa mudança na abordagem utilizada abre-se o caminho para os sistemas que existem atualmente.

Nos dias que correm, o reconhecimento ótico de caracteres é realizado por sistemas complexos, e que integram várias tecnologias. De uma forma geral, quase todos seguem uma arquitetura semelhante à apresentada na Figura 2.12 (Abdulkader 2009; Breuel 2008; Mori et al. 1992; Smith et al. 2009; Smith 2007):

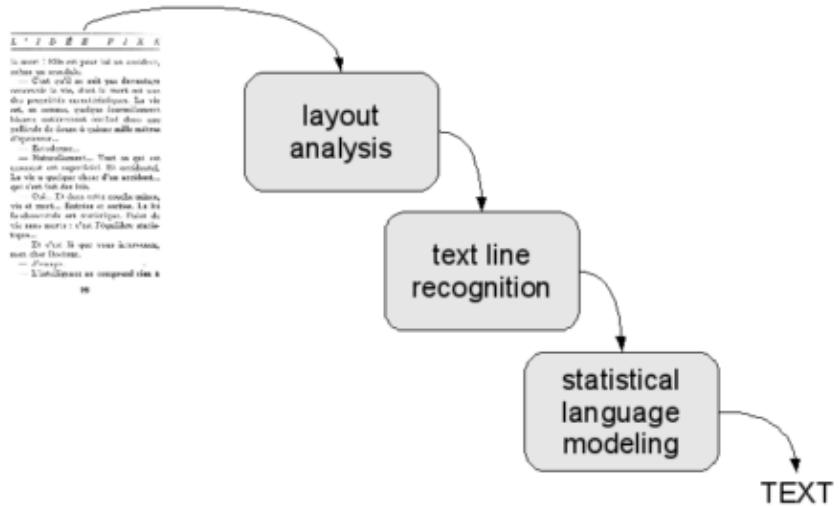


Figura 2.12 - Modelo de funcionamento de um OCR

O motor OCR utilizado nos sistemas gratuitos existentes no mercado é o Tesseract, que é um dos mecanismos pioneiros, conforme referido acima, e que teve um grande progresso qualitativo. Progresso este mais acentuado desde que há alguns anos foi aproveitado e melhorado pela Google. A sua estrutura de funcionamento é apresentada na Figura 2.13, e conforme se verifica, assemelha-se à arquitectura apresentada na Figura 2.12, que corresponde ao modelo de funcionamento de um OCR, em geral.

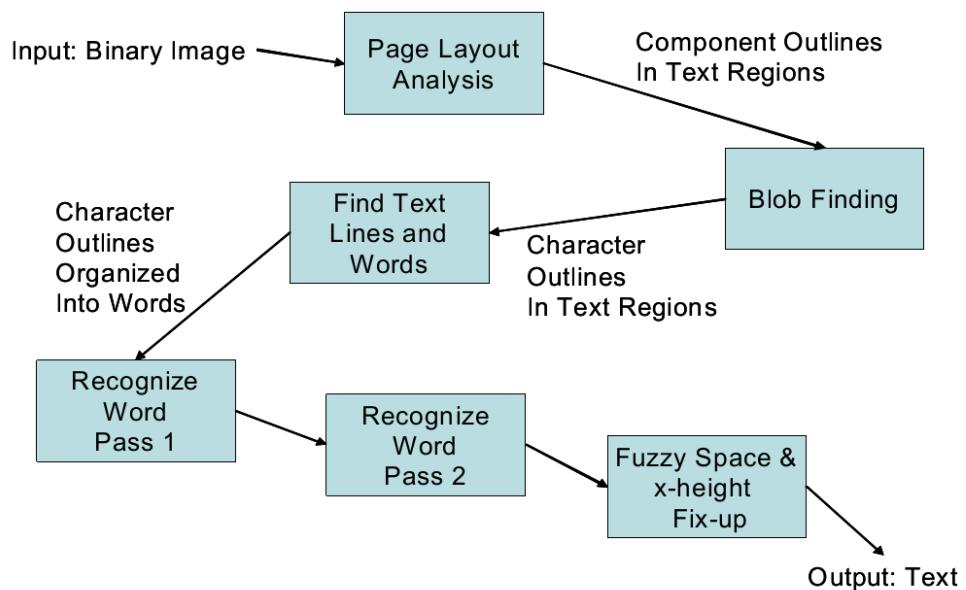


Figura 2.13- Estrutura de funcionamento do Tesseract (Smith et al. 2009)

2.4. Leitura automática de documentos

A leitura automática de documentos, refere-se a todo e qualquer processo, ou mecanismo, que permita de forma automatizada recolher a informação textual existente em imagens de documentos. Para cumprir este objetivo, são usadas várias formas, mas a mais em voga é a análise semântica e o recurso a bibliotecas de “conhecimento”.

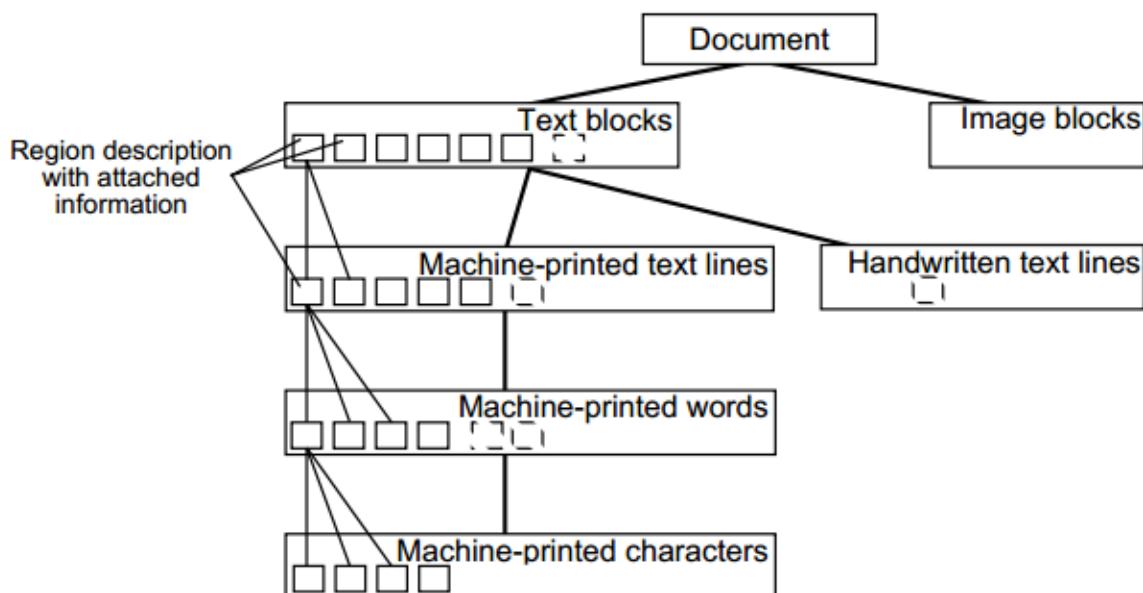


Figura 2.14 - Modelo genérico para definir a estrutura dum documento “(Kauniskangas 1999)”.

A análise semântica consiste na contextualização e percepção de texto após ter sido digitalizado através dum mecanismo OCR (Egozi et al. 2011; Radinsky et al. 2011; Deerwester et al. 1990; Lee et al. 2009). Existem várias formas de interpretar o texto, sendo as mais conhecidas e usadas a LSA (Latent Semantic Analysis), ESA (Explicit Semantic Analysis) e a TSA (Temporal Semantic Analysis). Para todos estes métodos, é necessário criar uma representação do texto, por exemplo, através de (BOW) bag-of-words representation, que consiste em considerar todos os termos reconhecidos como palavras-chave independentes e assim criar uma representação do texto. LSA é

uma técnica estatística que cria matrizes correlacionais entre palavra-documento, palavra-conceito e conceito-documento (Deerwester et al. 1990; Lee et al. 2009).

Uma das mais usadas na categorização de texto é a ESA, pois mantém o texto num formato perceptível, tratando-o como um conjunto de conceitos. Neste método, a um conceito está associado um vetor de palavras, e às palavras encontradas é lhes associado um peso, que é o nível de correspondência que cada palavra tem com o conceito. Ou seja, para cada conceito existem várias palavras e uma palavra pode estar associada a vários conceitos ao mesmo tempo, com pesos diferentes, conforme a sua correspondência ao mesmo, (Egozi et al. 2011).

Recorrendo-se à TSA o funcionamento é semelhante à ESA mas invertendo o sentido da ligação, significa que para cada palavra passam a existir vários conceitos e um mesmo conceito pode ser atribuído a várias palavras, contrariamente ao ESA, existindo aqui também um atributo que demonstra a precisão da correlação palavra-conceito. (Radinsky et al. 2011)

Conforme veremos no capítulo seguinte, a abordagem seguida neste trabalho vai ser diferente, pois vai ter em consideração as localizações e tipos de elementos presentes no documento, para detetar a respectiva semântica. Isto, pois a proposta destes autores não se adequa à documentação oficial e pessoal, que é um dos focos deste trabalho. Adicionalmente, o objectivo vai mais além da classificação de documentos, mais a leitura e interpretação de texto obtido dos documentos.

2.5. Reconhecimento de padrões

Conforme mencionado anteriormente, antes de se poder obter o conteúdo de um documento, é primeiro necessário identificar qual o documento específico que se está a processar. Nesta secção, apresentamos alguns métodos considerados adequados para a identificação de documentos. Cada uma destas técnicas permite, de uma ou de outra forma, identificar as características unívocas de cada documento, permitindo assim a sua identificação. Dada a abrangência deste tema, mencionam-se apenas alguns destes métodos, nomeadamente aqueles baseados em redes neurais e em Rough Sets, sendo esta última efetivamente utilizada no mecanismo de LAD desenvolvido.

2.5.1. Rough Sets

É um método utilizado no tratamento de informação, apresentado em 1982, por Pawlak (Pawlak 1982), que se baseia numa abordagem estatística, sobre dados imprecisos. Esta teoria assenta no conceito de indiscernibilidade entre objetos similares, assumindo que existem objetos com características de tal forma semelhantes que não é possível distingui-los. Pretende-se assim agregar todos os objectos com informação semelhante em classes, usando fundamentos matemáticos para fazer essa seleção (Øhrn 2000; Pawlak 1982; Pawlak 1997; Pawlak 2002; Pawlak & Skowron 2007; Radzikowska & Kerre 2002).

O processo catalogação/classificação começa por um conjunto de dados denominado de Sistema de Informação (SI), que contém vários objectos, que são as entradas horizontais de uma tabela de informação e os seus atributos, que são as colunas. Neste SI, distinguem-se duas classes disjuntas de atributos, os atributos condicionais e os de decisão, conseguindo-se assim definir uma tabela de decisão. Esta tabela é então usada como referência para se agruparem os objectos, formando assim conjuntos de objetos indiscerníveis entre si, sendo então estabelecidas Classes Equivalentes, o que leva à construção de uma nova tabela. Com as Classes Equivalentes identificadas é agora possível definir aproximações. Segundo Pawlak (Pawlak 1982; Pawlak 1995; Pawlak 1997) existe uma aproximação inferior, onde temos todas as classes equivalentes que certamente pertencem à classe de decisão em análise (Figura 2.15), que inclui o sub-conjunto (SET) dos exemplos positivos da tabela de decisão inicial. A aproximação superior, contém todas as classes que podem pertencer, ou não. Todas as outras classes que não estão incluídas em nenhuma das aproximações, definem-se como não pertencentes à classe de decisão.

Conforme descrito no capítulo seguinte, relativo à descrição do desenvolvimento do sistema LAD, esta técnica permite obter regras de inferência capazes de classificar a informação contida no SI, concretamente identificar a classe de cada documento lido através do “scanner”.

A aplicação de Rough Sets sobre uma tabela de decisão permite uma posterior geração de regras de decisão

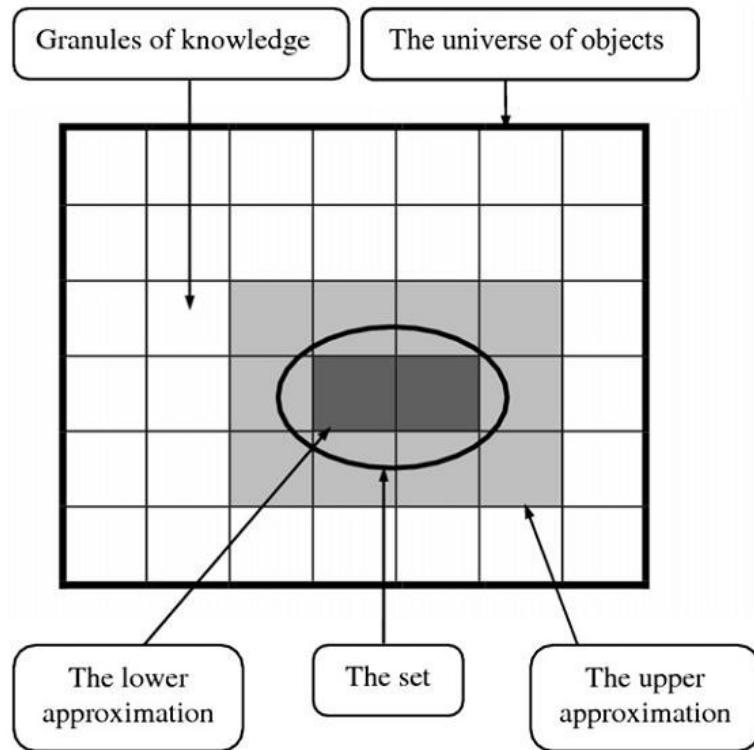


Figura 2.15 - Rough Set (Pawlak & Skowron 2007)

Para o processo de construção de regras, usa-se a Matriz de discernimento produzida anteriormente, que leva à obtenção de um Sistema de decisão com o qual se pode efetuar a previsão de resultados e classificação de classes.

A validade, precisão e abrangência são características importantes na tomada de decisão, visto que vão permitir verificar quais das respostas é a mais correta e que deve ser tomada em consideração. Em relação a este aspecto, a descrição de Rough Sets permite aferir a validade da informação de quantos objetos na Tabela de Decisão cumprem a condição. O parâmetro Precisão indica quais desses objetos obedecem à regra completa, e por fim a Abrangência, conforme o nome indica, fornece a relação entre o número de vezes que determinada regra se aplica e o total de objetos que tem o mesmo valor no atributo de decisão.

Todos estes cálculos estatísticos de certificação da validade de cada regra foram comprovados e demonstrados por Pawlak (Pawlak & Skowron 2007; Pawlak 2002; Pawlak 1997).

Resumidamente, RS é uma técnica usada para gerar regras para as quais é necessário um treino inicial. Esta técnica baseia-se em conceitos matemáticos mais concretamente probabilísticos os quais vão permitir identificar as regras mais credíveis e que devem ser valorizadas, podendo assim distinguir-se o que pode ser tido como verdadeiro, do que se deve tomar somente por possível.

2.5.2. Redes Neuronais

Trata-se de uma técnica motivada pelo reconhecimento do grande potencial existente em exemplos biológicos, especificamente a grande capacidade do cérebro humano para resolver problemas complexos de forma simples e rápida, bem como a forma de processar informação, completamente diferente da dos computadores. A plasticidade aparenta ser o que torna o cérebro numa máquina tão eficiente pois este adapta as suas redes de neurónios de forma a melhor resolver os problemas que lhe são apresentados. Um exemplo de uma rede neuronal ilustra-se na Figura 2.16.

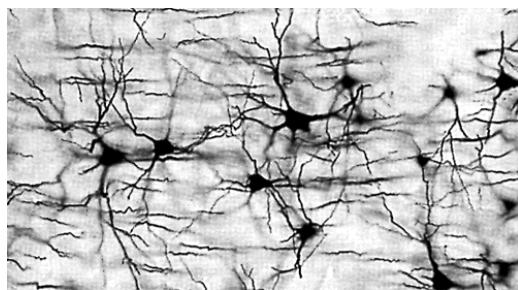


Figura 2.16 - Exemplo de Rede Neuronal no cérebro humano

A rede neuronal artificial ou rede neuronal como normalmente é conhecida (RN), trata-se de uma técnica de processamento paralelo de grande capacidade. A RN adquire conhecimento sobre o universo envolvente através de um treino, que resulta num correspondente ajustamento dos pesos sinápticos, ou seja, os pesos do relacionamento inter-neuronal (Haikin 1998; Ripley 2008), com os quais faz um mapeamento de entradas e saídas (Figura 2.17).

O modelo mais comum neste tipo de redes é o apresentado na figura 2.1.7, onde temos uma função transferência que alberga todos os valores de pesos inter-neuronais, os quais em seguida passam a entrada na função de ativação local onde é calculada a decisão final.

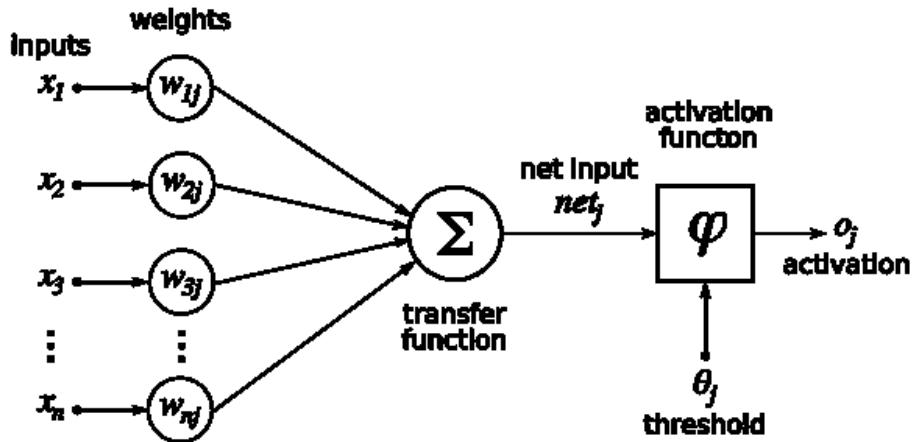


Figura 2.17 - Estrutura de uma Rede Neuronal (Chakraborty 2010)

Existem várias funções de ativação das RN, sendo as de uso mais comum a função Linear, a Sigmoide e a Threshold, que são representadas pelas seguintes expressões:

$$\text{Threshold } y = f(v) = \begin{cases} 1, & \text{se } v \geq 0, \\ 0, & \text{se } v < 0. \end{cases}$$

Equação 2

$$\text{Linear } y = f(v) = \begin{cases} 1, & \text{se } v \geq +\frac{1}{2}, \\ v, & \text{se } +\frac{1}{2} > v > -\frac{1}{2}, \\ 0, & \text{se } v \leq -\frac{1}{2}. \end{cases}$$

Equação 3

$$\text{Sigmóide } y = f(v) = \frac{1}{1 + e^{-av}}$$

Equação 4

Estas expressões ditam a forma como a rede reage às entradas e os resultados que se obtêm, o tipo de função deve ser selecionado conforme o objetivo pretendido (Chakraborty 2010). Sendo a Threshold a mais rígida, onde só são permitidos 2 valores; em oposição à Sigmoide, que é a mais flexível, permitindo um leque mais alargado de valores para o resultado final

O treino ou aprendizagem inicial, conforme também é conhecido, é fundamental para o funcionamento mais adequado das RN. Esta fase consiste em ensinar a rede a funcionar com determinado tipo de entradas. Isto é feito através da execução da RN, recorrendo a um lote de exemplos das diversas categorias que se pretende identificar. Usando um algoritmo de treino, os pesos inter-neuronais vão sendo reajustados de maneira a estarem preparados para permitir uma boa avaliação das entradas e correspondentes valores de saída.

Existem vários tipos de algoritmos de treino que podem ser aplicados, que se dividem em dois grandes grupos, os supervisionados e os não supervisionados, sendo as respectivas RN habitualmente denominadas de redes neuronais supervisionadas ou não supervisionadas como se pode ver na Figura 2.18 (Chakraborty 2010).

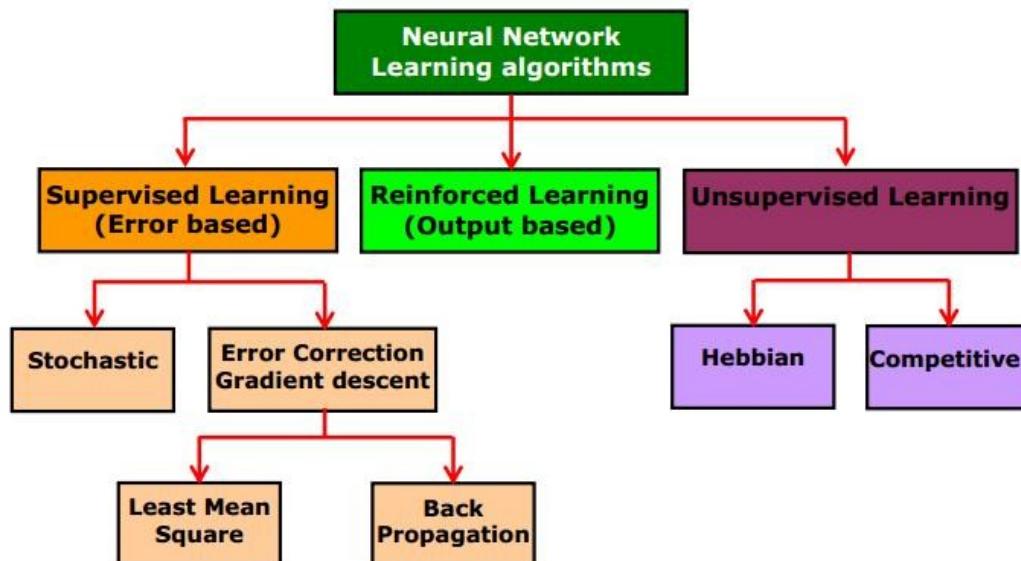


Figura 2.18 - Gráfico tipologias de Redes Neuronais

A aprendizagem supervisionada usa um conjunto de treino conforme o exemplo, onde cada um fornece um conjunto de entradas e saídas pretendidas (exemplo Tabela 2.1). Este tipo de treino permite uma maior fiabilidade da RN, pois a rede irá saber o que deve reconhecer. Neste tipo de aprendizagem o algoritmo mais comum, segundo (Haikin 1998), é o Back-Propagation (BP). A aprendizagem de reforço é um caso particular da aprendizagem supervisionada, embora também precise de um supervisor, neste tipo de aprendizagem são fornecidos exemplos de entradas, o algoritmo infere uma resposta e o supervisor só tem de classificar como certa ou errada o que vai servir de indicador para melhorar as respostas subsequentes.

Cores do Documento										
Tam_x	Tam_y	c1	c2	c3	c4	c5	c6	c7	c8	tipo
1004	637	16992	561	183	574760	0	6662	22120	18270	1
1004	637	16992	561	183	574760	0	6662	22120	18270	1
1004	637	16992	561	183	574760	0	6662	22120	18270	1
1232	845	33244	744	3912	652195	0	186956	162415	587	0

Tabela 2.1 - Exemplo de dados usados numa rede neuronal de teste

Caso se trate de RN não supervisionadas, só são fornecidos dados para simular entradas, o algoritmo de treino vai-se encarregar de tentar distinguir padrões existentes para identificar classes e treinar o sistema a reconhecer as mesmas, tal como referido anteriormente. Não é uma técnica tão fiável quando comparada com as RN supervisionadas, pois pode inferir classes que não existam, ou não descobrir categorias que deviam existir. A grande mais valia é a sua independência e o facto de ser auto-suficiente.

2.5.3. Redes ou Mapas de Kohonen

Trata-se de uma rede neuronal com características muito específicas e próprias, que lhe dão um grande ênfase e as distinguem de todas as outras, principalmente pelo facto de ser uma rede neuronal que se consegue auto-organizar. Possui também a especificidade de ser constituída em duas camadas (Kohonen 1990; Kohonen 2001; Haykin 1994; HAYKIN 2001), conforme ilustrado na Figura 2.19. Na literatura é também denominada de SOM (Self-Organized-Maps). Este tipo de redes não necessita de um algoritmo de treino supervisionado, porque faz autonomamente a avaliação dos pesos sinápticos.(Kohonen 1990; Kohonen 2001; Haykin 1994; HAYKIN 2001)

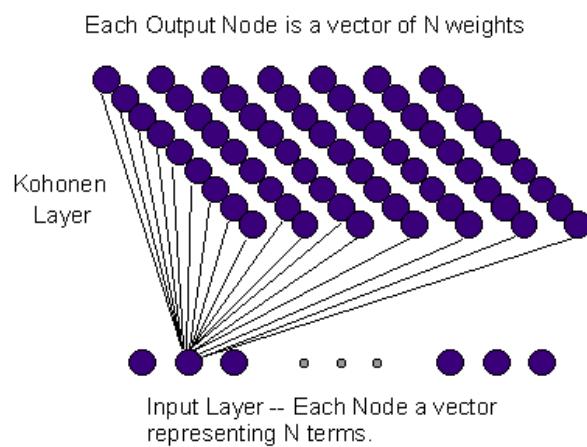


Figura 2.19 - Exemplo de Kohonen Neural Network

O nome de mapa de Kohonen vem da sua forma de funcionamento, visto que cada entrada é associada como que a uma posição no mapa e o algoritmo funciona como um mapeador dos atributos à entrada, que vai identificando o caminho pelos neurónios do "mapa" que provocam a ativação dos neurónios na camada de saída.

3. Desenvolvimento do sistema

LAD

3.1. Modelação do sistema

3.1.1. Ilustração do Conceito de LAD

A leitura automática de documentos consiste na aquisição de informação contida em documentos em suporte físico. Este processo é executado mediante a utilização de tecnologias de informação, nomeadamente leitores, scanners e software de reconhecimento de caracteres (definido como OCR). De forma a incorporar semântica ou interpretação nos documentos adquiridos, são também utilizados métodos que permitem observar um determinado documento como uma fonte estruturada de informação, permitindo identificar, interpretar ou estabelecer um significado à informação obtida. Por exemplo, a leitura dum documento de identificação para formato digital, permitirá não apenas o simples "scan" do documento, mas também obter o nome concreto da pessoa, o seu endereço, a naturalidade, entre outros. A captação dessa informação interpretada tem assim uma utilidade bastante mais significativa.

Este processo de leitura automática dum documento pode ser comparado, por similaridade ao processo de preenchimento duma GUI em que o utilizador sabe qual a informação que deverá ser introduzida em cada campo, que no final será armazenada numa base de dados, conforme ilustrado na Figura 3.1. Neste caso, o utilizador sabe onde colocar cada pedaço de informação relevante nessa GUI.

Dados do Proprietário do Veículo	
Nome:	<input type="text"/>
Endereço:	<input type="text"/>
Bairro:	<input type="text"/>
Estado:	<input type="text"/>
Cidade:	<input type="text"/>

Figura 3.1 - Exemplo de inserção de dados do proprietário de um veículo por um utilizador humano.

De forma similar, esta informação pode ser obtida a partir dum documento em suporte físico, sendo que desta vez os dados são obtidos através da leitura automática, conforme ilustrado na Figura 3.2



Figura 3.2 - Obtenção de informação através de LAD

Dado que através da LAD não existe intervenção humana, para que os documentos possam ser efetivamente lidos e interpretados, é necessário definir previamente os modelos que caracterizam estes documentos. Tipicamente, é necessário definir em que regiões ou segmentos do documento se encontra a informação necessária. Por exemplo, para o documento na Figura 3.3, é necessário definir as coordenadas das regiões correspondentes, aos pixéis dos campos de Data, endereços, ID, entre outros.

Form 23-EP Application for Enrollment to Practice Before the Internal Revenue Service as an Enrolled Retirement Plan Agent (ERPA) (Rev. January 2009)

Department of the Treasury Internal Revenue Service

See instructions on page 2.

Important things you need to do before you file this form:

- Take and pass the Enrolled Retirement Plan Agent Special Enrollment Examination (ERPA - SEE).
- Read Circular 230.
- Visit www.irs.gov to file and pay electronically or enclose a check or money order for \$125 made payable to the Internal Revenue Service.

This fee is non-refundable.

Check here if you are a former Internal Revenue Service employee and enter the date you separated from the Service 12 / 64 / 1980

Part 1: Tell us about yourself

1 Social Security Number 1 2 4 - 7 8 - 5 2 3 4

2 Print full legal name LYONS CHRISTOPHER

Left First

3 Current address 48 NELSON ROAD

Number Street Suite or room number

ALBANY NY 12201 UNITED STATES ZIP code 37485-1201

4 Enter the candidate number assigned to you by American Institute of Retirement Education, LLC. (AIRE) 37485-1201

5 Do you have an EIN No Yes If Yes, enter the Employer Identification Numbers (EINs) below

Enter the Employer Identification Numbers (EINs) below.

EIN	Name
5a	

For IRS use:
Enrollment Number
Date Enrolled:

- Extract Checkbox
- Extract Date
- Extract ID Numbers
- Extract Name
- Extract Address
- Extract Zip Code
- Extract State

Figura 3.3 - Exemplo de documento e campos a recolher (Rankl & Effing 2010)

Um exemplo dum documento que pode ser lido de acordo com esta abordagem ilustra-se na Figura 3.4



Figura 3.4 - Cartão do cidadão (Ministros, 2009)

O processo que permita efetuar a leitura de documentos e a respectiva conversão para um formato digital, conforme ilustrado na (Figura 3.5), mediante a utilização de métodos OCR. Basicamente, um documento é lido através de scan, e o texto respectivo é obtido através de OCR. Dependendo da localização do texto no documento, podemos classificar, reconhecer ou interpretar cada segmento do texto lido (por exemplo, o ID, o nome, etc).

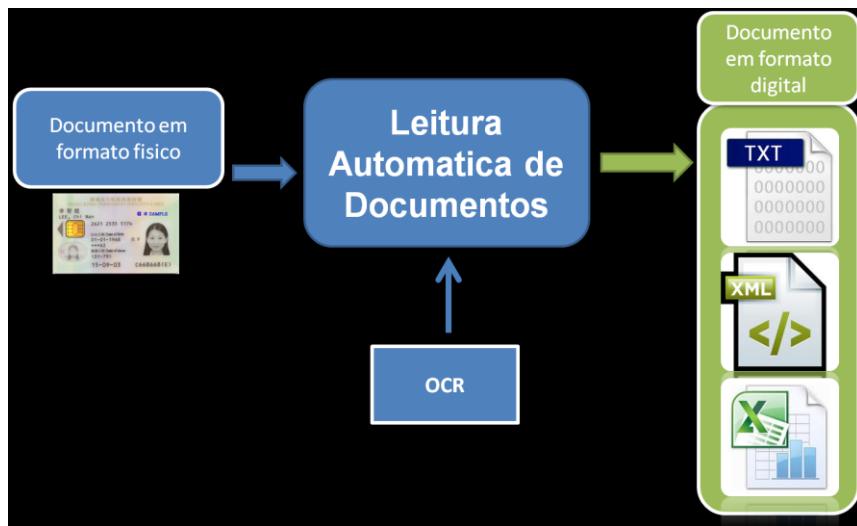


Figura 3.5 - Processo de Leitura Automática de Documentos

Dado existir uma variedade praticamente infinita de tipos de documentos, seria também útil que o sistema fizesse uma identificação prévia do tipo de documento, para assim conseguir identificar quais as zonas do documento que contêm texto, e melhor ainda, atribuir um significado a cada porção de texto reconhecido. Por exemplo, no caso de o sistema estar a ler um bilhete de identidade, ele poderia saber que uma zona do documento corresponde ao nome duma pessoa, outra zona à data de nascimento e outra à filiação. Desta forma poder-se-á obter a informação constante do documento da forma já “interpretada”, em que cada elemento de texto reconhecido seria acompanhado de uma etiqueta identificativa, conforme ilustrado na Figura 3.6. O resultado da leitura poderia assim ser feito na forma de um documento XML.



Figura 3.6 - Exemplo de recolha de informação de um documento de identificação

Nas secções seguintes deste capítulo, faz-se a descrição do sistema desenvolvido, conforme esboçado na Figura 3.2. Esta descrição começa pelo estabelecimento dos requisitos funcionais e não funcionais. Durante a especificação deste sistema, algumas componentes serão descritas utilizando uma notação gráfica, por exemplo UML ou diagramas de entidades e relacionamentos, conforme for mais adequado.

3.1.2. Requisitos Funcionais

Os requisitos funcionais identificados para o sistema LAD proposto são os que constam na tabela 3.1., cada linha da mesma possui um requisito que é acompanhado por uma descrição sumária.

Tabela 3.1- Requisitos funcionais para o sistema LAD

Requisitos Funcionais		
Requisitos	Descrição	Figura
RF1	Leitura de um Documento	Figura 3.7
RF2	Identificação manual do documento (pelo utilizador)	Figura 3.8
RF3	Identificação automática do documento	Figura 3.9
RF4	Leitura Interpretada de um Documento	Figura 3.10
RF5	Criação de <i>Modelos</i> para novos documentos	Figura 3.11
RF6	Criação automática de novos <i>Modelos</i>	Figura 3.12

Nas secções seguintes, descreve-se a especificação do sistema LAD, tendo em conta a satisfação desses requisitos. Por exemplo, o requisito RF3 foi satisfeito através da implementação de uma componente de identificação dos tipos dos documentos baseada em *Rough Sets*.

O requisito RF1 é o mais importante do sistema, estabelecendo que cada documento lido do seu suporte físico deverá ficar disponível em suporte digital (Figura 3.7).



Figura 3.7 - Processo de leitura dum documento em suporte físico

Permitir que o utilizador possa selecionar o documento que vai ser lido, a partir de um repositório de *modelos* já disponíveis, dessa forma o texto reconhecido poderia ser estruturado e interpretado, podendo o documento final ser em XML (Figura 3.8).

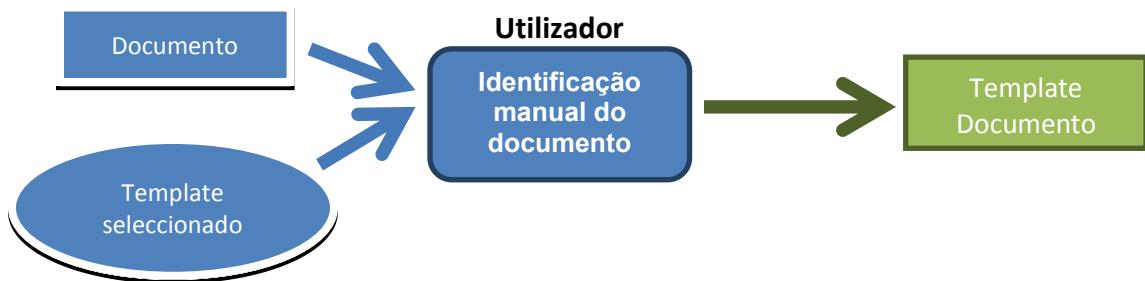


Figura 3.8 - Identificação manual dum documento

Este requisito é similar ao requisito RF2, mas em que o sistema identifica automaticamente qual é o documento a ser lido.

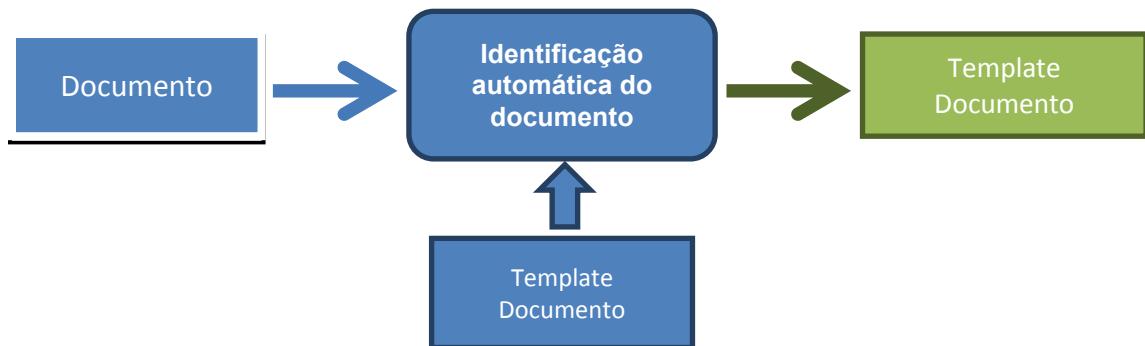


Figura 3.9 - Identificação automática dum documento

Requisito que recebe de entrada um documento em suporte físico, converte para digital e recolhe o seu texto interpretado.

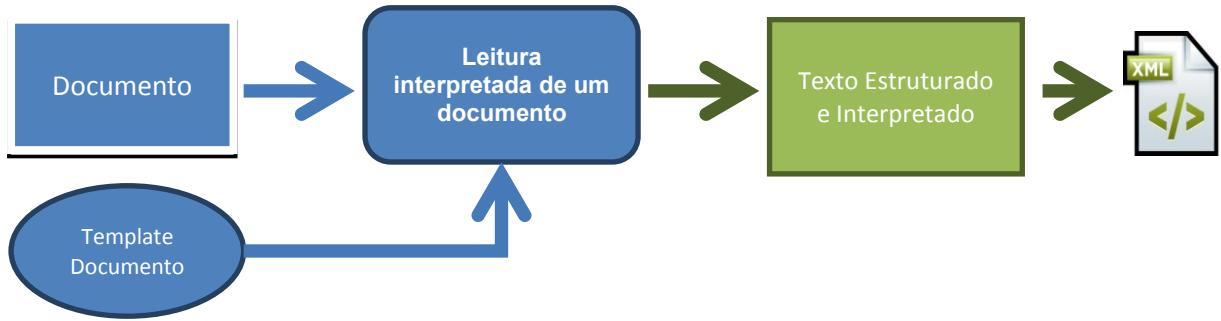


Figura 3.10 - Processo de leitura interpretada dum documento

Com o requisito seguinte, garante-se a possibilidade de aprender a reconhecer novos tipos de documento através da criação de novos *modelos*, criados manualmente por um utilizador através da seleção dos campos relevantes e do seu significado.

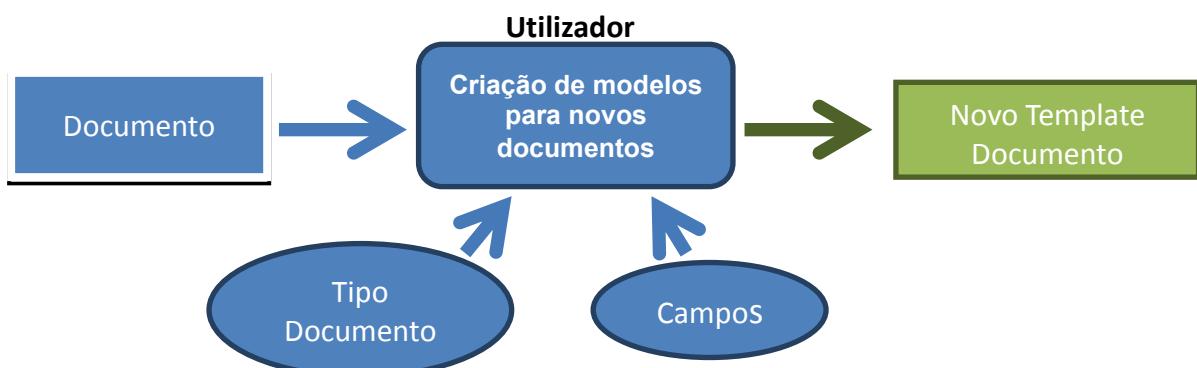


Figura 3.11 - Processo de criação manual dum novo modelo

À semelhança do RF5, permite criar um modelo de documento para reconhecimento futuro, mas neste caso o processo de seleção de campos é feito de forma automática.

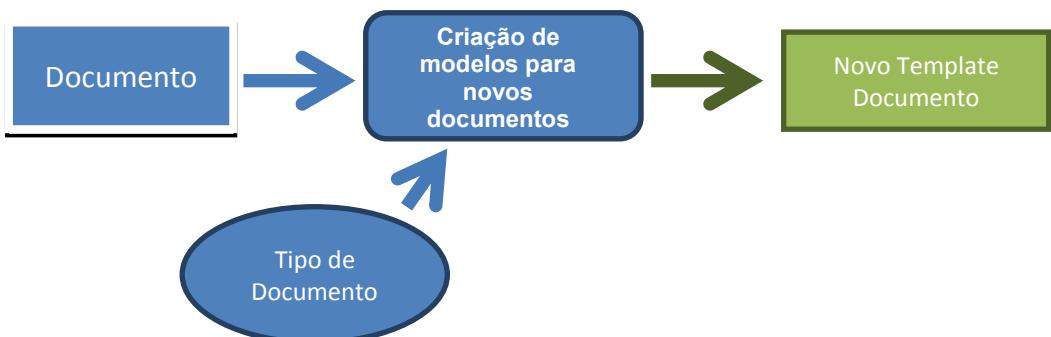


Figura 3.12 – Processo de criação automática de um novo modelo

3.1.3. Requisitos Não Funcionais

Tipicamente, os requisitos não funcionais estabelecem o nível adequado de qualidade para cada função ou requisito funcional que o sistema irá executar. Exemplos de requisitos não funcionais genéricos são: tempo de resposta, escalabilidade, entre outros. No caso do sistema LAD, a tabela 3.2 apresenta um conjunto mais específico de requisitos não funcionais, adequados para este caso.

Tabela 3.2 - Requisitos Não Funcionais

Requisitos Não Funcionais	
Requisitos	Descrição
RNF1	Tempo de leitura e conversão do documento Este requisito influencia na utilidade deste sistema, se aplicado em contexto real visto definir o tempo de execução do mesmo.
RNF2	Qualidade da conversão Dado que o estado atual da tecnologia OCR, que reconhece texto sempre com alguma margem de erro, pretende-se reduzir a quantidade de erros de conversão para um valor considerado aceitável
RNF3	Adaptabilidade Requisito que garante se o sistema consegue identificar novos documentos, nomeadamente mediante a capacidade para poder considerar novos modelos de documentos.
RNF4	Escalável Ao longo do tempo, o sistema pode aceitar uma maior quantidade de novos tipos de documentos.

3.2. Especificação do Sistema de Leitura Automática de Documentos

De forma a poder-se descrever adequadamente cada componente do sistema, é necessário proceder a uma respectiva descrição, dentro dum nível de formalismo adequado, facilitando uma especificação mais rigorosa e com menos ambiguidades.

Desta forma, iremos definir os conceitos de documento, tipo de documento, modelo e funcionalidade LAD, seguindo uma notação formal.

Definição 3.1 (Documento em suporte físico) – Um documento em suporte físico (*DSF*) é um documento que contém texto, podendo também conter imagens. Cada *DSF* pode ser representado por um tuplo $dsf = (d_i, imagem_i)$. Daqui em diante considere-se a existência do conjunto de documentos em suporte físico $DSF = \{dsf_1, dsf_2, \dots, dsf_n\}$ com $1 \leq i \leq n$, em que cada dsf_i representa o identificador de cada documento. Por esta definição, um *DSF* constitui uma representação do documento físico.

Definição 3.2 (Reconhecimento ótico de caracteres) – O reconhecimento ótico de caracteres (OCR) pode ser definido através do operador abstrato $OCR: DSF \rightarrow \text{Texto}$, que aceita como argumento um documento em suporte físico e fornece o respectivo texto (através da operação “*scan*”).

Definição 3.3 (Documento em suporte digital) – Um documento em suporte digital (*DSD*) é um documento que foi criado mediante a utilização de OCR (definição 3.2) sobre um *DSF* (definição 3.1). Daqui em diante considera-se a existência do conjunto dos documentos em suporte digital:

$$DSD = \{(d_i, txt_i) \mid d_i \in DSF \wedge txt_i = OCR(d_i)\}$$

A definição seguinte permite definir os modelos dos documentos, conforme introduzidos na secção 2.4. Basicamente é necessário identificar os segmentos dum documento, cujo texto tenha interesse em reconhecer, conforme ilustrado na Figura 3.13.

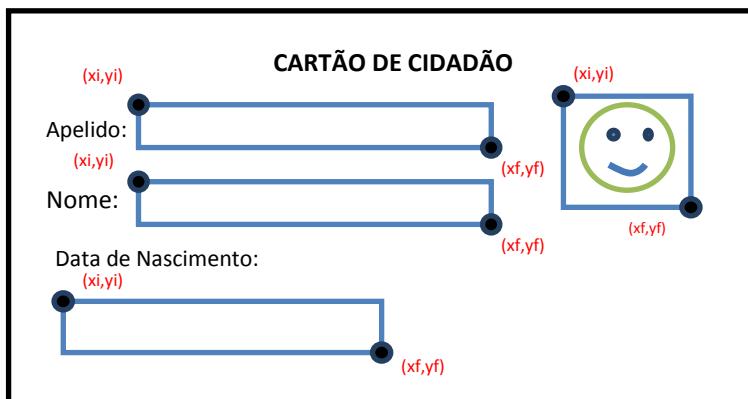


Figura 3.13 - Definição do modelo de um documento

Definição 3.4 (Modelo dum documento) – Um modelo dum documento, pode ser definido pelo tuplo $DM=(t_i, S)$ em que t_i representa um identificador, ou *label*, que identifica um tipo de documento (ex: BI nacional), e em que S_i representa um conjunto definido como:

$$S = \{(Label_j, x_{ij}, y_{ij}, x_{fj}, y_{fj}) \mid Label \in Strings, x_{ij}, y_{ij}, x_{fj}, y_{fj} \in \mathbb{R}\}$$

Este conjunto permite descrever quais os segmentos (S), ou zonas, de um tipo de documento que contém texto útil e qual o seu significado, descriminado pelas respectivas labels. Daqui em diante, considere-se a existência do conjunto “Document Model Set” $DMS =\{DM_1, DM_2, \dots\}$, ou dito doutra forma, $DMS =\{(t_1, S_1), (t_2, S_2), \dots\}$, que contém todos os modelos definidos no sistema LAD. O conjunto Strings é o conjunto de todas as palavras/Labels.

Definição 3.5 (Documento em suporte digital interpretado) - Trata-se de um documento que foi lido da sua forma física, e em cada segmento de texto lido foi interpretado, recorrendo a um modelo de documento adequado (conforme a Definição 3.4). Pode ser definido pelo tuplo (d_i, T_λ) em que d_i representa o identificador de um documento e T_λ é o conjunto:

$$T_\lambda = \{(Label_1, Text_1), (Label_2, Text_2), \dots\}$$

Cada *Label* de T_λ está definida no respectivo modelo do documento (definição 3.4), ficando assim associada a uma semântica concreta (ex: nome, endereço, localidade, telefone, etc). Considere-se doravante a existência do conjunto dos Documentos em suporte digital interpretados $DSDI = \{(d_1, T\lambda_1), (d_2, T\lambda_2), \dots\}$. O conteúdo de T_λ pode mais tarde ser convertido num documento XML, com utilização posterior noutros processos dum sistema.

Definição 3.6 (Leitura interpretado dum documento) – A leitura interpretada dum documento (LID) é definida pelo operador que recebe um documento em suporte físico, o seu respectivo modelo e devolve o documento em suporte digital interpretado. A sua assinatura é então

$$LID: DSF \times DMS \rightarrow DSDI$$

Definição 3.7 (Identificador modelo documento) – Consiste na operação de identificação do modelo $m_i \in DMS$, que é necessário para a leitura interpretada dum documento em suporte físico $d_j \in DSF$, conforme a Definição 3.1 e Definição 3.4, é abstratamente definida através do operador $imd: DSF \rightarrow DTS$.

Definição 3.8 (Mecanismo de leitura automática dum documento) – O mecanismo para a leitura automática dum documento é abstratamente definido através da seguinte álgebra:

$$S = \{DSF, DSD, DSDI, DMS, Text, Strings, \mathbb{R}\}$$

$$\Omega = \{ocr, lid, imd, scan\}$$

$$\Sigma_{\Omega} = \{ \begin{cases} \text{ocr:DSF} & \rightarrow \text{Text} \\ \text{lid:DSD} \times \text{DTS} & \rightarrow \text{DSDI} \\ \text{imd: DSF} & \rightarrow \text{DTS} \\ \text{scan:} & \rightarrow \text{DSF} \end{cases}$$

O conjunto S agrupa os tipos ou classes estabelecidos nas definições anteriores. O conjunto Ω representa os operadores considerados nas mesmas definições.

De uma forma abstrata, utilizando as assinaturas presentes na definição 3.8, a leitura interpretada dum documento em suporte físico $d_i \in DSF$, convertida num documento interpretado em suporte digital $DSDI_j$ pode ser modelada através das *queries* $d_i = \text{scan}$, e $d_{\text{dsdi}_j} = \text{lid}(d_i, \text{imd}(\text{ocr}(\text{scan})))$. Nesta querie, e para evitar repetir-se operações, o processo de scanner só é feito uma vez.

Por exemplo, o resultado da leitura interpretada dum documento BI, com esta query seria

$$dsdi_{bi} = \{(\text{nome}, 'antonio'), (\text{endereço}, 'Lisboa'), \dots \}$$

3.3. Implementação do sistema LAD

3.3.1. Arquitetura do sistema LAD

Nesta secção são apresentados e detalhados todos os elementos que englobam o sistema LAD construído, o seu modo de funcionamento e a sua localização na arquitetura.

Na Figura 3.14 representa-se a arquitetura global da ferramenta desenvolvida, constituindo uma visão global e como tal de alto nível. Ao longo desta secção descreve-se a estratégia seguida na implementação de cada um desses blocos.



Figura 3.14 - Arquitetura global do sistema

A arquitetura apresentada resume o rumo inicialmente definido para este projeto, que consistiu na criação de uma ferramenta, que fornecendo-lhe um documento, o processe, verifique a sua identidade, valide a mesma, e por fim lhe aplique um motor OCR recolhendo a informação textual interpretada. Para tal, são necessários 4 blocos como se verifica na Figura 3.14, um para realizar a conversão do documento físico para um formato digital, primeiramente uma imagem digital, um outro bloco, definido como Identificador que autonomamente realiza o reconhecimento de qual o tipo de documento de identificação a ser analisado. Posteriormente, é validada pelo bloco da Base de Dados com os modelos de documentos, o qual seleciona de acordo com a tipologia do documento as zonas sobre as quais será aplicado o motor OCR. O bloco final trata-se do motor OCR, nas zonas previamente selecionadas faz a conversão do texto numa imagem digital para texto em formato digital.

Existe um bloco que não foi apresentado na imagem, pois não influencia o processo principal do software que se trata de um elemento que caso não seja identificado o tipo de documento automaticamente, conforme ilustrado na Figura 3.15, permite a construção de um novo modelo para identificações futuras.

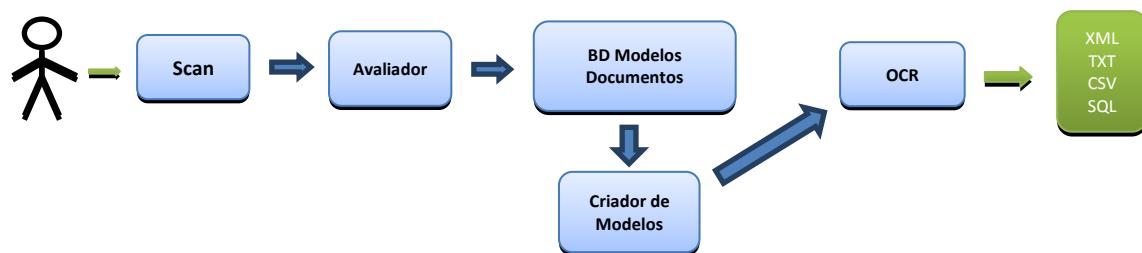


Figura 3.15 - Arquitetura do sistema quando não é identificado documento

De seguida, apresenta-se uma descrição mais pormenorizada dos blocos apresentados nas Figura 3.14 e Figura 3.15, começando pelo Scan, o qual pode ser definido por dois sub-blocos. O primeiro consiste no scanner que representa o processo concreto de criação de uma imagem digital do documento inserido no equipamento scanner, recorrendo para isso à impressão do reflexo obtido após a projeção de luz contra o documento físico num sensor de imagem (Figura 3.16).

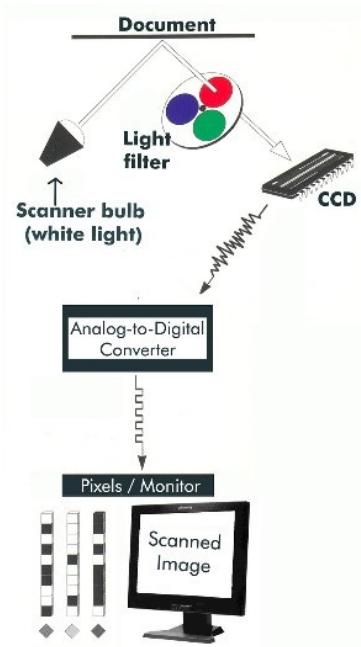


Figura 3.16- Processo mais comum de digitalização de imagem

Após a digitalização do documento obtém-se uma imagem digital do mesmo, a qual não se encontra ainda nas condições adequadas para a sua avaliação, identificação, interpretação e leitura estruturada, sendo para tal necessário aplicar algumas modificações na imagem as quais são executadas pelo Processador Imagem, correspondendo esta parte ao segundo sub-bloco, conforme ilustrado na Figura 3.17.

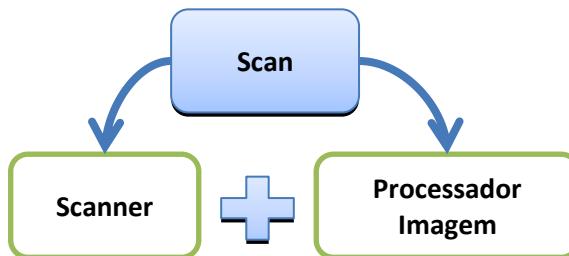


Figura 3.17 – Processos que definem o bloco Scan

O Processador Imagem faz uma verificação inicial para identificar se existe ou não necessidade de aplicar um corte na imagem, para só incluir o documento após esta verificação. É então aplicado uma conversão no regime de cores da imagem, passando esta de RGB (a cores) para preto e branco. Nesta altura, a imagem encontra-se nas condições adequadas para se começar a recolher informações, processo este que é iniciado ainda neste bloco. É então aplicado um algoritmo de componentes ligados e de Haar Cascade Classifier, ou seja, de identificação de faces, algoritmos estes

que já foram descritos no capítulo anterior deste documento. Daqui se obtêm os elementos necessários para arrancar com o Avaliador.

O Identificador, conforme o nome indica, avalia a imagem do documento e tenta verificar e autenticar a sua identidade. Para isso, são executados dois processos distintos, um primeiro que ao receber os elementos recolhidos pelo Processador Imagem faz uma triagem inicial, onde são avaliados alguns elementos iniciais. Sendo os restantes elementos avaliados pelo segundo processo, baseado na aplicação das regras de Rough Sets.

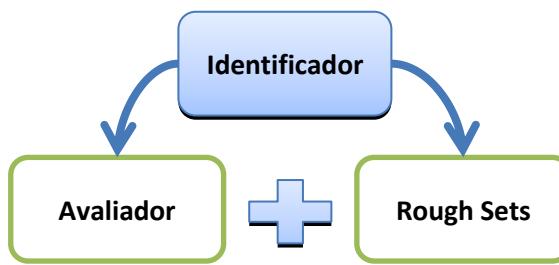


Figura 3.18 – Processos que definem o bloco Avaliador

Os elementos usados por este avaliador inicial, são as dimensões do documento, e com o algoritmo *haar cascade classifier*, a existência ou não de faces no documento e respectiva localização. Faz-se então uma comparação destes dados com os modelos existentes na base de dados de modelos, que contêm todos os tipos de documentos identificáveis e conhecidos ate ao momento. Caso exista alguma correspondência são então avaliados os restantes dados recebidos e com os quais é feita uma confirmação do resultado desta avaliação inicial e em alguns casos melhorar os resultados obtidos, estreitando o leque de hipóteses finais a só um elemento. Caso não seja possível identificar nenhuma correspondência e dado como encerrado o processo de identificação, pode ser iniciado um processo de criação de um novo modelo de documento, para que seja possível futuramente identificar documentos do mesmo tipo do recolhido, e obter a sua informação de forma estruturada.

Em ambos os casos de sucesso numa identificação ou criação de um novo modelo, é utilizado o algoritmo de Rough Sets. Num caso, para confirmação da identificação, enquanto que para outro usa-se para treinar o software para confirmações futuras.

A noção de Rough Sets já foi introduzida neste documento num capítulo anterior, por isso de seguida, será feita uma descrição do algoritmo recorrendo aos elementos

que foram utilizados neste projeto, para um melhor enquadramento com os objetivos do mesmo. Este algoritmo é usado sempre que se treina o sistema inicialmente ou após a criação de novos modelos de documentos, para que seja possível a sua identificação. Rough Sets trata-se de “uma Framework matemática para a indução de regras”(Hvidsten 2010), regras estas que permitem fazer a classificação e identificação de categorias/classes distintas.

3.3.2. Leitura (OCR) e interpretação de documentos

Para o desenvolvimento da ferramenta pretendida nesta dissertação, foi necessário estabelecer algumas estruturas de dados, tanto de características referentes à imagem do documento, como para identificação do tipo de documento que se trata e para a construção de modelos de tipos de documentos.

A estrutura inicial consistia num Doc, o qual sabe-se que contém texto e está associado a um tipo de documento. O tipo vai ter correspondência num Modelo de Doc, que guarda os atributos identificativos do mesmo, ou seja, o próprio Tipo a que está associado, os campos de interesse nesses Doc's , o tamanho médio da imagem produzida com esse tipo de doc's e por fim um array de características. Cada Campo, é identificado pelo nome do campo, que define também do que se trata e a sua localização no documento (Figura 3.19). Esta estrutura segue as definições introduzidas na secção 3.2, com mais atributos contextuais, que naturalmente surgem durante a passagem de uma especificação abstracta para um modelo concreto.

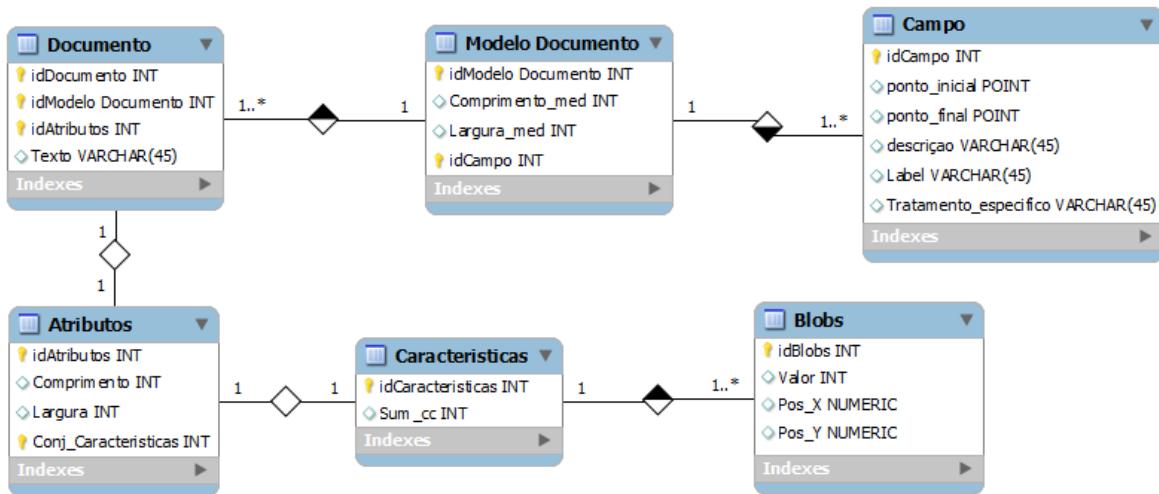


Figura 3.19 - Modelo Inicial Sistema LAD

Após testado, verificou-se que a estrutura de dados do Modelo de Doc's não conseguia atribuir singularidade suficiente aos vários tipos existentes. Como tal, não permitia a sua eficiente identificação, constatando-se assim a necessidade de inserir mais informação referente aos Modelos e consequentemente a criação de uma nova classe. A classe Face (Figura 3.20), que consiste na existência ou não de uma cara no doc e o seu posicionamento, foi a escolhida conseguindo-se assim com este novo atributo aumentar a eficiência do algoritmo.

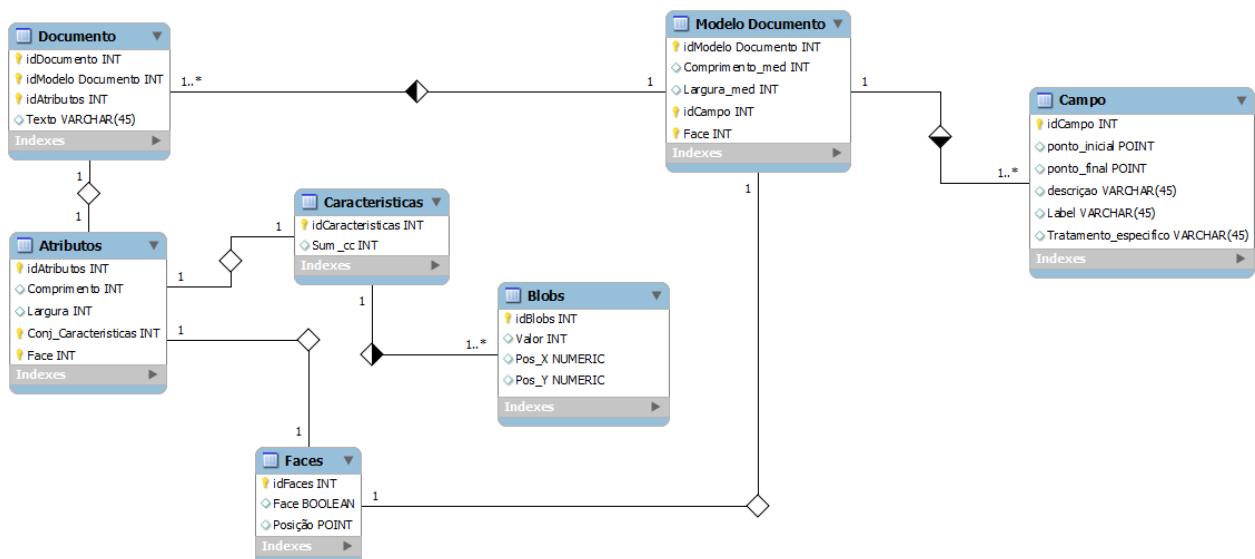


Figura 3.20 - Modelo Final Sistema LAD

Quando é feita referência às características da imagem dos documentos, trata-se de uma matriz 3x3, na qual é contabilizado o número de blobs no quadrante respectivo da imagem. Blob é a nomenclatura usada para bloco de componentes ligados, e conforme descrito anteriormente, refere-se a um conjunto de pixéis que de alguma forma têm contacto com outros pixéis, formando assim os blocos de componentes ligados.

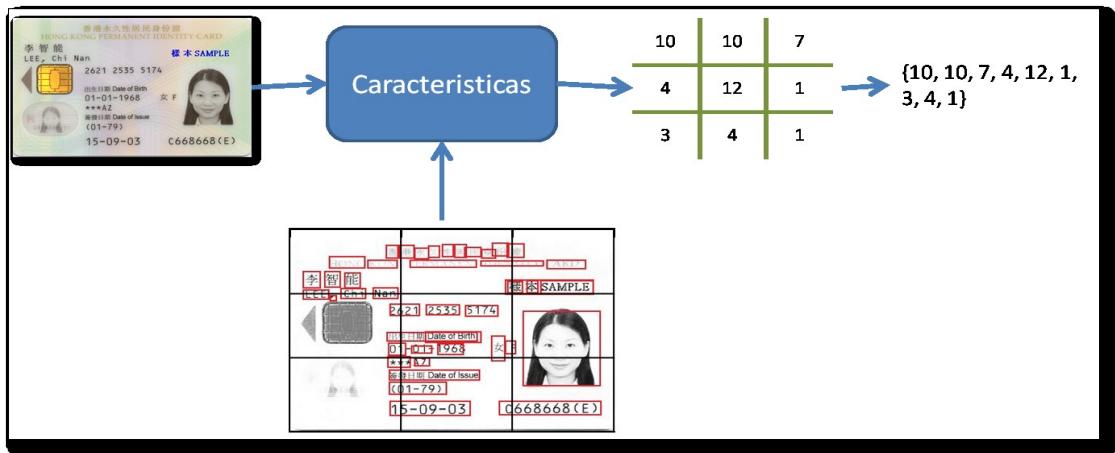


Figura 3.21 - Processo de recolha de características do documento

As características servem para identificar quais as zonas do doc com mais e menos informação textual. Daí ser aplicado um limite de tamanho para os blocos de forma a garantir que só é identificado texto e assim excluir-se qualquer imagem, como por exemplo, uma bandeira, uma cara ou um símbolo de determinada entidade.

Na identificação de faces, faz-se recurso ao já explicado anteriormente, denominado de Classificador em cascata de Haar. Pretende-se obter, caso exista, o posicionamento de Caras nas imagens de doc. Posteriormente as características e a Face usam-se para fazer a classificação das imagens e assim identificar o seu tipo.

Para facilitar esta classificação é também utilizado o tamanho da imagem como filtro inicial.



Figura 3.22 - Processamento de imagem aplicando Haar-like features

A sequência de eventos que levam à identificação e respectiva recolha do conteúdo do documento inicia-se com a introdução de um documento (no caso específico usado em laboratório, documentos de identificação) no scanner. Daí em diante o processo é automático, primeiro a imagem do documento, que é criada pelo scanner, é pré-processada de forma a só conter a área do documento e verifica que o doc está alinhado. De seguida, são recolhidas as características e a Face, incluindo-se também neste conjunto de características as dimensões da imagem. A partir destes dois últimos atributos (Faces e Dimensões) e considerando a lista de *modelos* existentes, é feita uma filtragem dos tipos possíveis, delegando para o RS a seleção final e ou a confirmação do resultado obtido.

O diagrama de sequência apresentado na Figura 3.23 ilustra de que forma as diversas componentes do sistema LAD interagem para satisfazerem os requisitos funcionais. O utilizador coloca o documento e procede à sua aquisição para formato digital (Requisito Funcional 1), a partir daí, procede-se à identificação do modelo de documento adequado (R.F. 2 e R.F. 3) e à aquisição das respectivas regiões que contêm o texto, obtendo-se assim informação interpretada (R.F.4).

Mais em baixo, na Figura 3.23, ilustra-se também o processo de criação de novos modelos de documentos (R.F. 5 e R.F. 6).

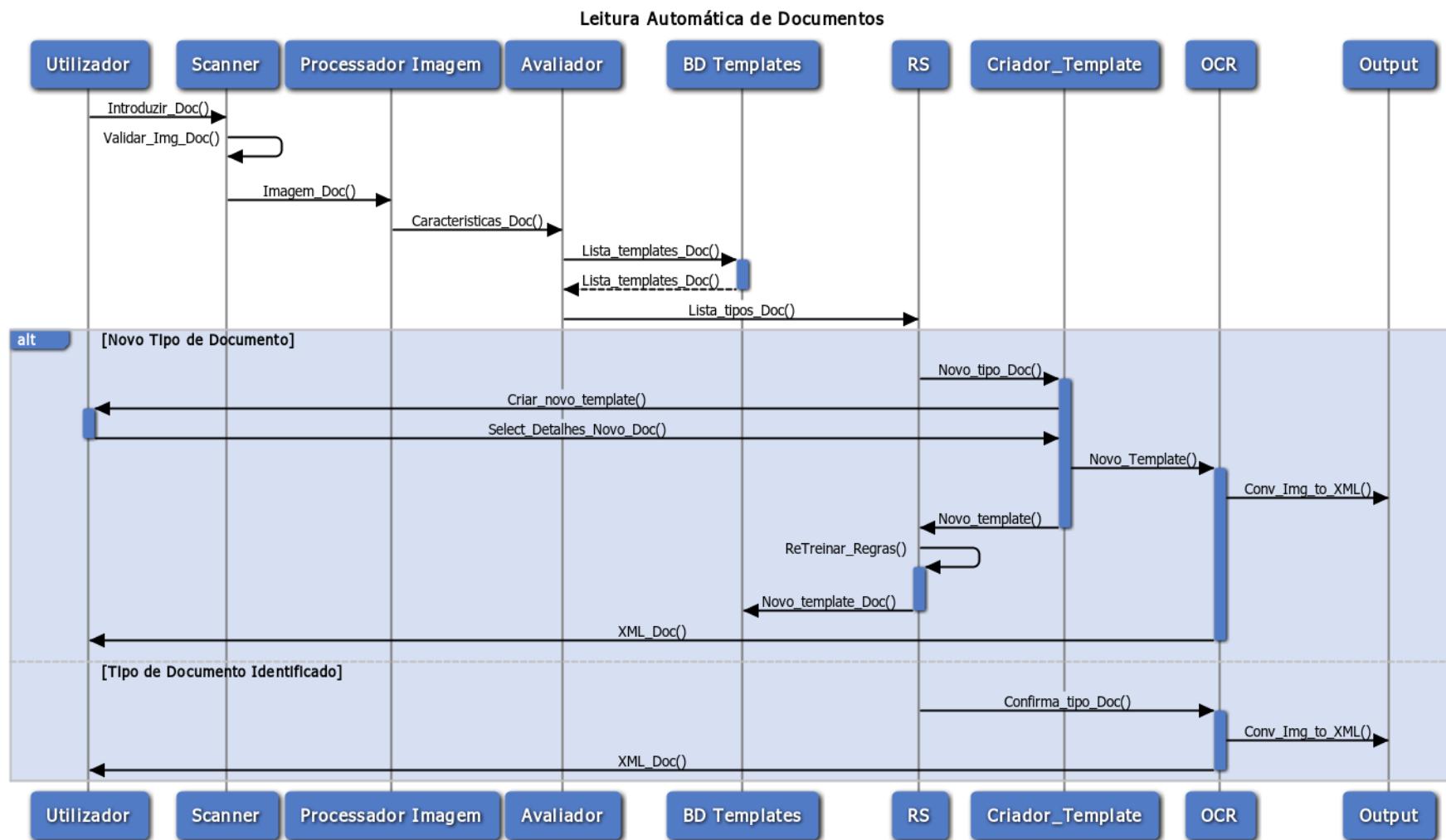


Figura 3.23 - Diagrama sequência UML do processo de funcionamento do software

3.3.3. Mecanismo de identificação dos modelos baseado em Rough Sets

A noção de Rough Sets já foi introduzida anteriormente, por isso de seguida será feita uma descrição da aplicação deste método numa identificação dos tipos de documentos. Este algoritmo é usado sempre que se treina o sistema, inicialmente ou após a criação de novos modelos de documentos, para que seja possível a sua identificação.

Tal como enunciado no capítulo 2, o processo de construção de regras RS, denominado de processo de treino, começa com um sistema de informação que contém várias entradas onde cada linha corresponde a um objecto e cada coluna está associada a um atributo, que no caso concreto do software LAD são as características recolhidas do documento digitalizado. Quando estes dados se fazem acompanhar por um atributo de decisão (última coluna da Tabela 3.3), ou seja, a identificação a que categoria/classe correspondem, passa a denominar-se de tabela de decisão. Em seguida, descreve-se a aplicação deste método numa tabela de decisão, que é utilizada na construção das regras de RS aplicadas no sistema LAD.

Tam_x	Tam_y	C1	C2	C3	C4	C5	C6	C7	C8	C9	Face	Face_x	Face_y	Tipo
1004	641	78	144	71	41	120	34	22	81	19	1	0	1	2
1232	974	120	203	92	68	145	44	45	91	34	0	0	0	0
1232	845	154	238	105	102	170	47	60	98	50	0	0	0	0
1228	974	196	296	128	120	196	50	76	106	74	0	0	0	0
1224	844	229	332	140	154	218	52	87	111	85	0	0	0	0
1216	941	272	390	163	182	255	60	100	118	106	0	0	0	0
1008	641	280	441	180	189	302	79	110	168	110	1	0	1	2
1004	637	294	470	180	199	320	82	117	193	114	1	2	2	1
1004	637	308	499	180	209	338	85	124	218	118	1	2	2	1
1004	638	321	552	191	221	380	103	137	253	122	1	0	1	2
1004	626	330	604	210	228	425	123	146	303	126	1	0	1	2
1008	626	352	659	216	239	468	140	158	338	130	1	0	1	2
1004	634	361	710	234	247	503	155	164	379	134	1	0	1	2

Tabela 3.3 - Conjunto de treino

A Tabela 3.3 é um conjunto de treino de RS. Por ter toda a informação possível, pode conter demasiada informação, podendo alguma ser redundante. Por isso, antes de se avançar no processo, é aplicada uma redução à tabela de decisão para se verificar se todos os seus atributos são necessários. A verificação processa-se da seguinte forma, é selecionada uma função booleana que verifica, atributo a atributo a sua "utilidade", comparando-o com os diferentes objetos e verificando se este atributo os permite distinguir, criando-se desta forma uma tabela de Decisão Tabela 3.4.

Objectos	Atributos Condicionais														Atributo de Decisão
	Documento	Tam_x	Tam_y	C1	C2	C3	C4	C5	C6	C7	C8	C9	Face	Face_x	Face_y
D1	1004	641	78	144	71	41	120	34	22	81	19	1	0	1	2
D2	1232	974	120	203	92	68	145	44	45	91	34	0	0	0	0
D3	1232	845	154	238	105	102	170	47	60	98	50	0	0	0	0
D4	1228	974	196	296	128	120	196	50	76	106	74	0	0	0	0
D5	1224	844	229	332	140	154	218	52	87	111	85	0	0	0	0
D6	1216	941	272	390	163	182	255	60	100	118	106	0	0	0	0
D7	1008	641	280	441	180	189	302	79	110	168	110	1	0	1	2
D8	1004	637	294	470	180	199	320	82	117	193	114	1	2	2	1
D9	1004	637	308	499	180	209	338	85	124	218	118	1	2	2	1
D10	1004	638	321	552	191	221	380	103	137	253	122	1	0	1	2
D11	1004	626	330	604	210	228	425	123	146	303	126	1	0	1	2
D12	1008	626	352	659	216	239	468	140	158	338	130	1	0	1	2
D13	1004	634	361	710	234	247	503	155	164	379	134	1	0	1	2

Tabela 3.4 - Tabela de decisão

A partir da Tabela 3.4, é possível começar a aplicar a teoria de Pawlak e procurar objetos indiscerníveis entre si, levando à definição de classes equivalentes, as quais englobam em si os conjuntos de objetos indistinguíveis. Na Tabela 3.5, tem-se a tabela de classes equivalentes para o sistema LAD, a partir da qual se constrói uma matriz de discernibilidade, para se identificar mais facilmente as funções de discernibilidade que conduzem à criação das regras de inferência para determinação dos tipos de documentos.

Classes Equivalentes	Tam_x	Tam_y	C1	C2	C3	C4	C5	C6	C7	C8	C9	Face	Face_x	Face_y	Tipo
E1={D1}	1004	641	78	144	71	41	120	34	22	81	19	1	0	1	2
E2={D2,D3}	1232	974	120	203	92	68	145	44	45	91	34	0	0	0	0
E3={D4,D5}	1228	974	196	296	128	120	196	50	76	106	74	0	0	0	0
E4={D6 }	1216	941	272	390	163	182	255	60	100	118	106	0	0	0	0
E5={D7,D10}	1008	641	280	441	180	189	302	79	110	168	110	1	0	1	2
E6={D8,D9}	1004	637	294	470	180	199	320	82	117	193	114	1	2	2	1
E7={D11,D12,D13}	1004	626	330	604	210	228	425	123	146	303	126	1	0	1	2

Tabela 3.5 - Classes equivalentes

Com a matriz de discernibilidade, conforme se verifica na Tabela 3.6 seguinte, apresenta-se uma matriz com a relação entre as várias classes equivalentes, identificando-se que atributos diferem entre si. Efectua-se essa averiguação entre classes equivalentes que não partilhem o mesmo atributo de decisão, conforme se pode verificar pelos campos rasurados na matriz, através da qual é possível de forma mais direta, simples (e intuitiva) construir as funções de discernibilidade, ou seja, as funções que permitem distinguir as classes equivalentes entre si e as próprias classes

existentes. Para se obter as funções referidas basta recolher, coluna a coluna e linha a linha, todos os conjuntos de atributos que permitem distinguir a classe a que corresponde essa coluna, de todas as outras.

	E1	E2	E3	E4	E5	E6	E7
E1							
E2	Tam_x,Tam_y,C1,C2,Face_e,Face_y						
E3	Tam_x,Tam_y,C1,C2,C3,C4,C7,C9,Face,Face_y						
E4	Tam_x,Tam_y,C1,C2,C3,C4,C5,C7,C9,Face,Face_y						
E5		Tam_x,Tam_y,C1,C2,C3,C4,C5,C7,C8,C9,Face,Face_x	Tam_x,Tam_y,C1,C2,C3,C4,C5,C7,C8,C9,Face,Face_y	Tam_x,Tam_y,Face,Face_y			
E6	C1,C2,C3,C4,C5,C6,C7,C8,C9,Face_x,Face_y	Tam_x,Tam_y,C2,C3,C4,C5,C7,C8,C9,Face,Face_x,Face_y	Tam_x,Tam_y,C1,C2,C3,C4,C5,C7,C8,C9,Face,Face_x,Face_y	Tam_x,Tam_y,C8,Face,Face_x,Face_y	Face_x,Face_y		
E7		Tam_x,Tam_y,C1,C2,C3,C4,C5,C6,C7,C8,C9,Face,Face_y	Tam_x,Tam_y,C1,C2,C4,C5,C6,C7,C8,C9,Face,Face_y	Tam_x,Tam_y,C2,C5,C6,C8,Face,Face_y		C2,C6,C8,Face_x,Face_y	

Tabela 3.6 - Matriz de discirnibilidade

Na posse das funções de discirnibilidade, pode dizer-se que já se tem acesso às regras de inferência de RS, denominadas de regras de decisão. Antes da transformação concreta em regras, estas funções são simplificadas, de forma a manter o menor número de atributos possível.

Após simplificadas, as funções de discirnibilidade são o que define cada regra. As regras são constituídas por duas partes, uma onde se identifica que atributos são necessários para identificar a classe; e na outra a classe que estes permitem distinguir das restantes. Em seguida, temos alguns exemplos das regras obtidas ao longo do projeto. No anexo deste documento apresenta-se a tabela de regras de inferência do sistema LAD.

Cada regra, conforme já foi explicado em capítulos anteriores, tem associada uma certa probabilidade estatística de ser verosímil. Visto tratar-se de um método baseado numa abordagem estatística, cada regra possui uma percentagem de precisão, abrangência e suporte. O suporte, representa o número de vezes que no sistema de decisão determinada regra é verdade. Dividindo este valor pelo número total de aparições no SD, da parte entre o IF e o THEN, obtém-se a precisão que, conforme o

nome indica, reflete a probabilidade de o resultado ser exato e não poder divergir. A abrangência calcula-se dividindo o suporte pela totalidade de ocorrências da classe identificada nesta regra no SD (Tabela 3.7).

Regras	LHS Support	RHS Support	RHS Accuracy	LHS Coverage	RHS Coverage	RHS Stability	LHS Length	RHS Length
CC7(22) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC7(45) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC7(60) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC7(76) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC7(87) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC7(100) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC7(110) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC7(117) => Tipo(1)	1	1	1.0	0.076923	0.5	1.0	1	1
CC7(124) => Tipo(1)	1	1	1.0	0.076923	0.5	1.0	1	1
CC7(137) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC7(146) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC7(158) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC7(164) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC9(19) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC9(34) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC9(50) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC9(74) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC9(85) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC9(106) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC9(110) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC9(114) => Tipo(1)	1	1	1.0	0.076923	0.5	1.0	1	1
CC9(118) => Tipo(1)	1	1	1.0	0.076923	0.5	1.0	1	1
CC9(122) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1

Tabela 3.7 - Lista parcial das regras RS e as estatísticas associadas

3.4. Arquitetura de Software

O projeto descrito neste documento inclui a criação de um software, sendo então necessário descrever a arquitetura física do sistema. Visto ter-se verificado a necessidade de testar várias ferramentas e algoritmos, é necessário utilizar uma linguagem de programação de base C# que permita uma fácil integração entre componentes. A Figura 3.24 apresenta as tecnologias a integrar e de que forma deve ser feita essa integração, se dentro de um mesmo bloco funcional ou através de comunicação entre si.

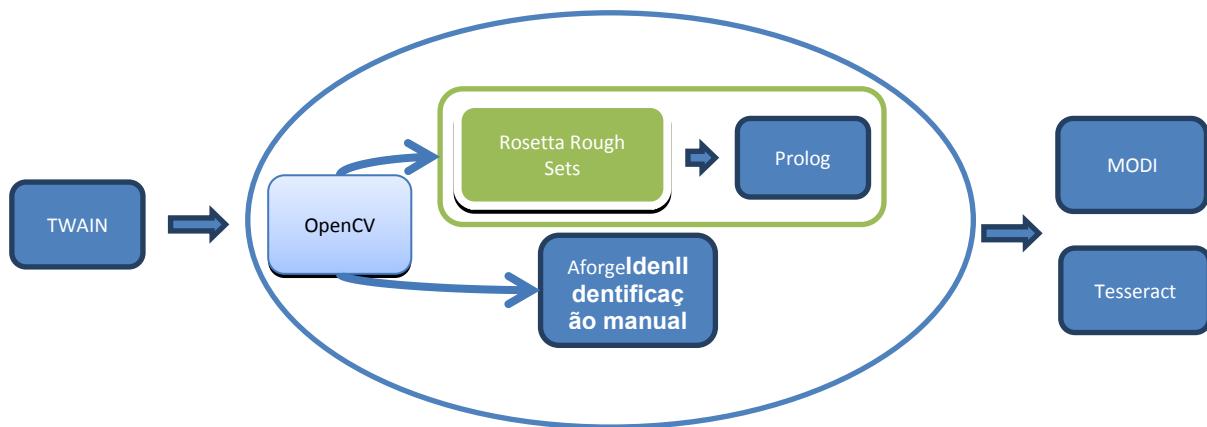


Figura 3.24 - Visão global da arquitetura e bibliotecas usadas para a implementar

Na secção seguinte descreve-se mais em detalhe a funcionalidade de cada um destes blocos e a forma como se integram e interagem dentro do sistema LAD.

3.4.1. Aquisição de imagem e pré-tratamento

Twain, trata-se de um padrão que tenta ser o mais universal possível, para permitir a integração entre equipamentos de aquisição de imagem e software. Facilita assim a utilização de diversos equipamentos, sem necessidade de qualquer software específico de um determinado fabricante. Para tal existe um conjunto de drivers para os diversos equipamentos englobados por este padrão. Sem a utilização destes drivers, a aquisição de imagens seria um processo bastante mais complexo. Daí ter-se optado pela sua utilização no projecto como ponte entre o scanner que faz a digitalização dos documentos e o software desenvolvido.

O OpenCV, conforme já referido anteriormente, é um elemento muito utilizado e útil para o tratamento de imagem digital. É uma biblioteca muito completa de funcionalidades na área do tratamento digital de imagens, embora contendo também algumas funcionalidades na área da inteligência artificial, conforme ilustrado na Figura 3.25.

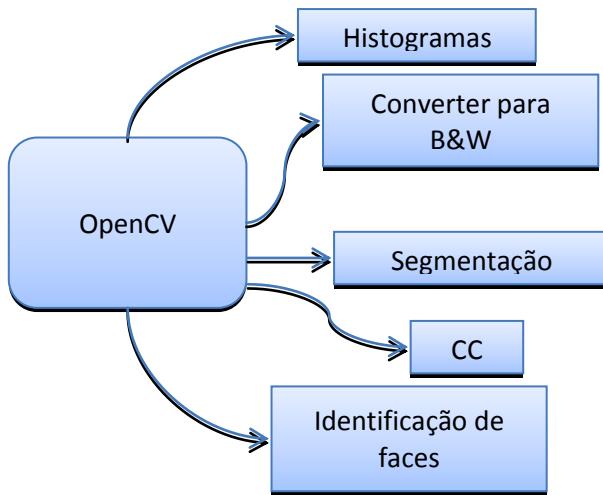


Figura 3.25 - Funções OpenCV usadas

Neste projecto, foi utilizada somente a vertente de tratamento de imagem, do OpenCV, que permite efectuar todos os pré processamentos da imagem, antes desta ser identificada, como se pode ver na Figura 3.25 através das funções utilizadas.

Em relação à biblioteca Aforge (Figura 3.26) a sua utilização deve-se principalmente à sua funcionalidade de componentes ligados, usada neste projeto aquando da recolha de características da imagem do documento e de forma secundária pelas suas funções no âmbito da inteligência artificial.

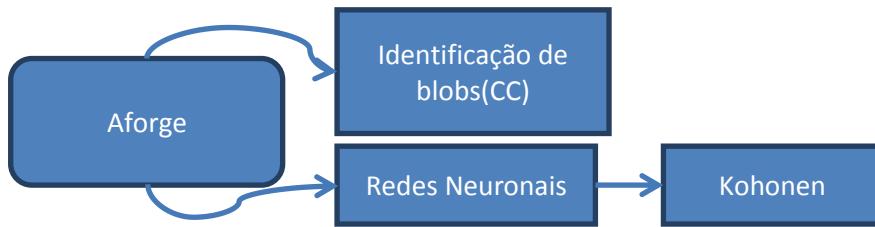


Figura 3.26 - Funções Aforge usadas

Especificamente para este projecto, utilizaram-se as funções de construção e treino de redes neurais nas suas diversas tipologias e arquiteturas; as quais foram usadas no software produzido, para testar a possibilidade de utilização na identificação de categorias substituindo os Rough Sets. A biblioteca Aforge, à semelhança da biblioteca OpenCV, é uma biblioteca muito diversificada, contendo funcionalidades de tratamento de imagem, inteligência artificial, robótica e tratamento de áudio, entre

outras. É uma biblioteca muito focada na integração com .Net, um aspecto importante no desenvolvimento do sistema LAD.

Para a construção dos Rough Sets, seu treino e criação de regras, utilizou-se uma ferramenta chamada Rosetta Rough Sets, que engloba todas as funções necessárias à criação dos sistemas de informação e seu tratamento até à criação de regras, conforme Figura 3.27.

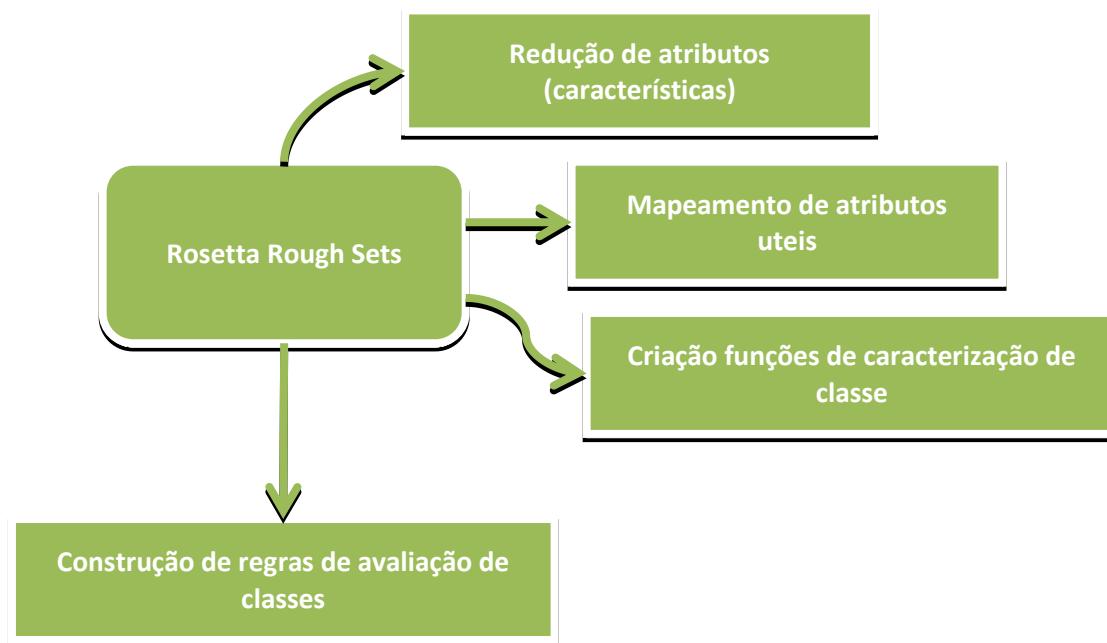


Figura 3.27 - Funções desenvolvidas com apoio da ferramenta Rosetta

Uma das grandes vantagens desta ferramenta é a possibilidade de importação e exportação em diversos formatos. Esta capacidade permite uma maior possibilidade de integração, a qual foi utilizada, tendo como dados de entrada para os RS um ficheiro Excel produzido pelo software e tendo como saída regras em Prolog, as quais podem ser usadas diretamente na ferramenta desenvolvida sem ser necessário qualquer adaptação ou conversão. Para que tal seja possível, insere-se uma “dynamic link library” (DLL), com código que permita ligação com as bibliotecas de Prolog do SWI-Prolog (Figura 3.28).

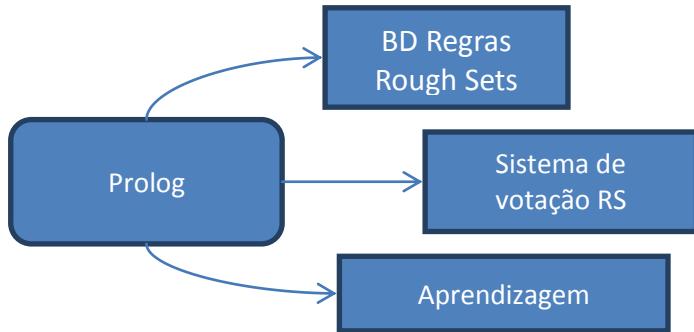


Figura 3.28 - Funcionalidades usadas recorrendo a Prolog

A linguagem de programação Prolog é direcionada para a utilização com uma linguagem natural e que vem sendo aprimorada para objetivos específicos ao longo dos tempos, sendo atualmente utilizada no desenvolvimento de soluções no ramo da inteligência artificial.

Neste projeto, o Prolog serve de ponte entre a ferramenta de Rough Sets e o sistema LAD, pois as regras produzidas pela ferramenta são em Prolog, ou seja, sempre que é necessário reconfigurar as regras quando se cria um novo modelo, as regras em Prolog são novamente recarregadas pelo sistema. A utilização do Prolog deveu-se à sua simplicidade de desenvolvimento e por facilitar a reconstrução de regras RS, permitindo assim uma mais rápida atualização do sistema para novos modelos de documentos, bem como a reutilização da ferramenta de Rough Set para a criação das mesmas.

Todos os blocos descritos integram-se recorrendo ao desenvolvimento do software em linguagem de programação .Net mais especificamente C#, dada a sua versatilidade de integração com bibliotecas e tecnologias (conforme ilustrado na Figura 3.29).

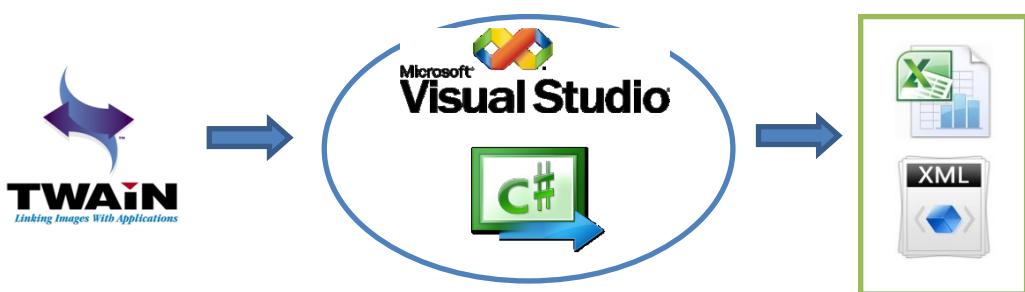


Figura 3.29 - Plataformas tecnológicas usadas

3.4.2. Equipamento utilizado no desenvolvimento

Os equipamentos usados para o desenvolvimento deste projeto foram um scanner e um computador (Figura 3.30), que serviu para o desenvolvimento e para simular uma estação de trabalho que recorra ao software desenvolvido.



Figura 3.30 - Ambiente de desenvolvimento montado para a implementação do software

Para a escolha do scanner a utilizar, foi feito um estudo inicial para tentar encontrar o que melhor se adequasse aos objetivos do projeto e futuramente a uma possível comercialização do mesmo. Foi então escolhido o scanner Fujitsu f-60 (Figura 3.31) devido às suas dimensões, visto pretender-se digitalizar somente documentos de identificação. Foram também tidas em conta a sua velocidade de execução, para garantir a celeridade de todo o processo e todas as funcionalidades extra associadas a este equipamento, sendo algumas delas muito úteis neste projeto, nomeadamente o recorte da zona útil e o alinhamento das imagens.



Figura 3.31 - Scanner utilizado no projecto

As necessidades computacionais deste projeto não são muito exigentes, pois o elemento que vai requerer mais disponibilidade do computador é o processo de classificação. Como os algoritmos escolhidos não necessitam de muitos recursos, por isso não existe a necessidade de um computador muito potente, o que é útil caso se pretenda transportar este projeto para uma vertente mais comercial, facilitando a sua implementação independentemente da infraestrutura do cliente.

3.5. Sistema LAD desenvolvido

O software construído possui uma interface gráfica que encontra-se aqui exemplificada nas figuras que se seguem (Figura 3.32, Figura 3.33, Figura 3.34 e Figura 3.35). A Figura 3.32 começa por apresentar o layout global do sistema LAD.

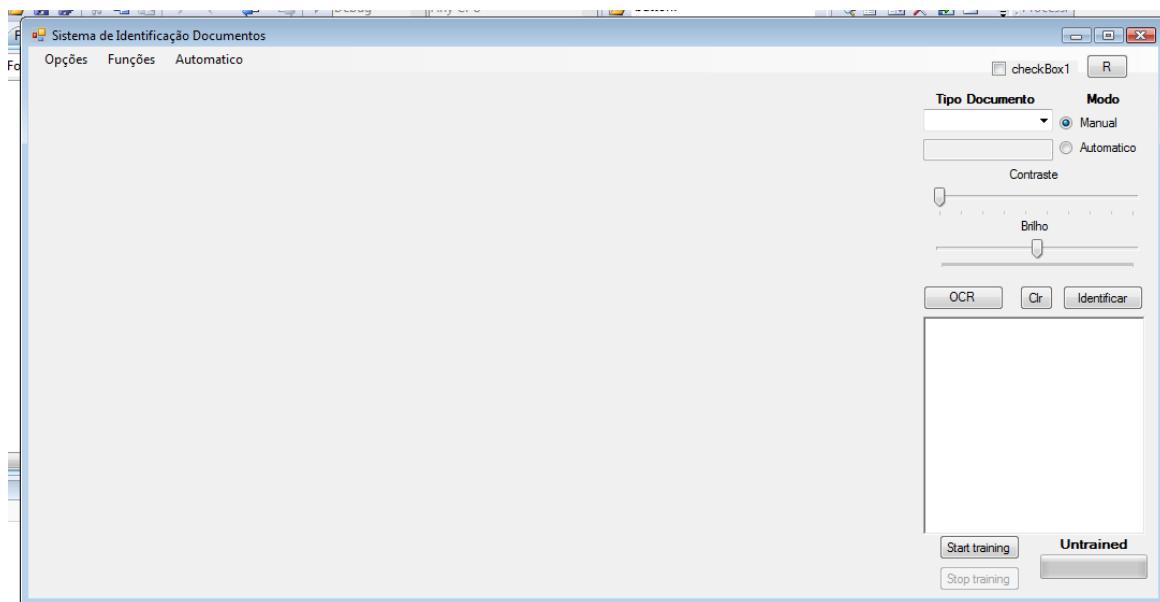


Figura 3.32 - Layout global da ferramenta

As diversas funcionalidades desenvolvidas ilustram-se na Figura 3.33. Conforme se pode constatar, os diversos menus de opções permitem aceder às funcionalidades estabelecidas durante o levantamento dos requisitos funcionais.

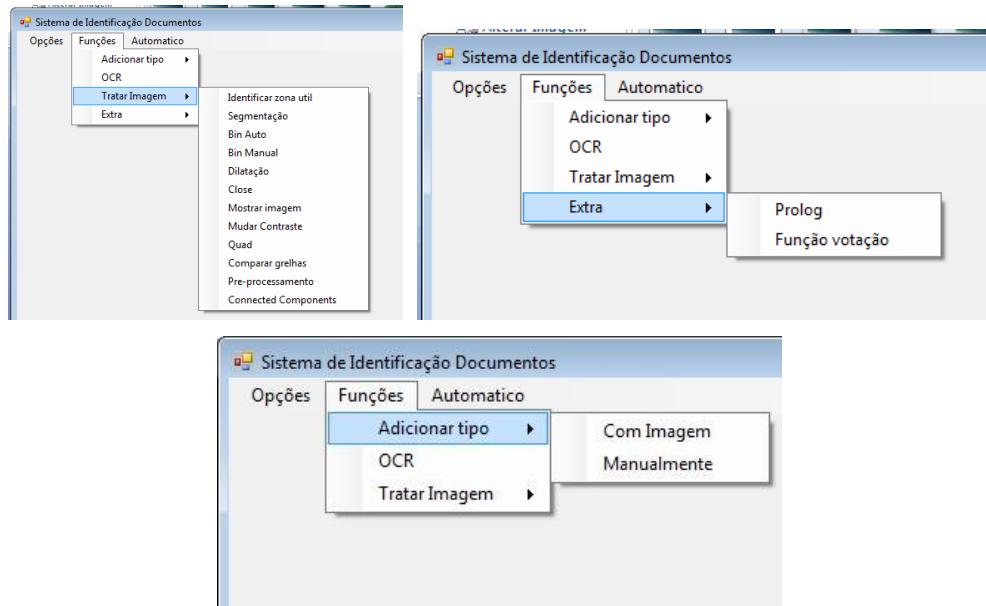


Figura 3.33 - Menus de funcionalidades do software

Conforme referido anteriormente, as funcionalidades que permitem a leitura automática de um documento são a digitalização do documento, o pré-tratamento da imagem digital obtida, que inclui binarização e as alterações de contraste e de brilho da imagem. Durante este passo, é também executada em simultâneo uma rotina que faz a recolha das dimensões da imagem, a sua segmentação, identificação de componentes ligados, identificação e localização de faces em imagens digitais. A partir destas informações, o software procede à identificação de documentos através de Rought Sets e Redes Neuronais. Desenvolveu-se também a funcionalidade de treino de redes neurais, aplicação de OCR a uma imagem digital ou sobre partes de uma imagem recorrendo 2 tipos de bibliotecas MODI e Tesseract.

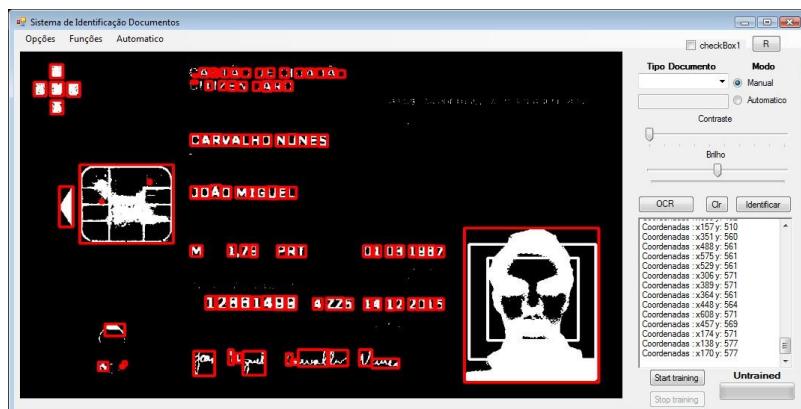


Figura 3.34 - Exemplo de aplicação de CC recorrendo à ferramenta desenvolvida



Figura 3.35 - Exemplo de leitura executada sobre um documento de identificação com o software criado durante este projeto

A forma como a ferramenta foi desenvolvida, permite que esta seja facilmente reajustável permitindo a substituição de cada bloco por um novo que se ache mais útil, e ou com melhor desempenho. O bloco de identificação de documentos é completamente independente do OCR, tal como o OCR é dos modelos de documentos, o que os junta é um núcleo central que faz a integração dos vários blocos e cria um processo único como se de um só bloco se tratasse Figura 3.36.

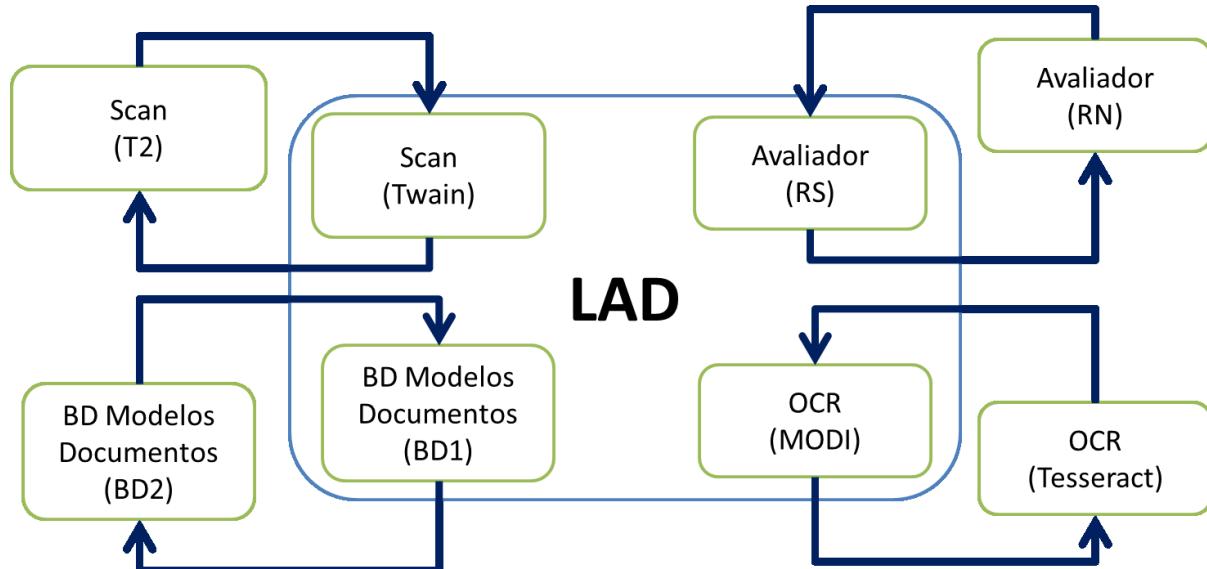


Figura 3.36 - Arquitetura em que cada componente pode ser substituída

3.6. Testes e Validação

Nesta secção vai-se analisar o desempenho da ferramenta desenvolvida, para assim poder verificar-se a sua implementação tendo em conta a correspondente especificação. Pode-se também certificar a sua validade, tendo em conta os requisitos funcionais.

3.6.1. Verificação

'A verificação dum sistema é efectuada para garantir que a sua implementação representa de forma precisa o modelo conceptual estabelecido. Efectuar a verificação ajuda a garantir que o modelo, os algoritmos e as funções foram implementados corretamente e que o modelo não contém erros ou quaisquer bugs.

Com a verificação, garante-se que a especificação da solução é completa e que não existem erros após a implementação. Contudo, não assegura que o modelo resolve determinado problema, tão somente atestando que segue as especificações do modelo, ou reflete corretamente o funcionamento num ambiente real' (B.H.Thacker et al. 2004).

Para verificação do software criado realizou-se um exemplo de leitura de um documento de identificação e foi-se confirmado, passo a passo, se o processo de LAD correspondia às especificações, previamente estabelecidas.

O processo de Leitura Automática de Documentos para ser iniciado requer um pré requisito fundamental, o documento a identificar, o qual se encontra em suporte físico, como o exemplo da Figura 3.37, começando logo neste passo por se demonstrar a utilização da Definição 3.1 (Documento em suporte físico), onde é descrito no que consiste um documento em suporte físico.



Figura 3.37 - Documento em suporte físico

Após cumprido o único pré requisito, pode-se avançar para o processo LAD, em concreto, inserindo o documento num scanner que converte o mesmo para um formato digital. Também este passo pode ser descrito por uma das definições enunciadas anteriormente, mais especificamente Definição 3.3 (Documento em suporte digital), que apresenta uma noção do que se deve considerar um documento em suporte digital aqui exemplificado com a digitalização do documento da Figura 3.37 que se concretiza na Figura 3.38



Figura 3.38 - Documento em suporte digital

Após digitalizado o documento, é necessário ser identificado, usando para isso os processos já referenciados neste capítulo num sub-capítulo anterior. Tal processo é tido em consideração na Definição 3.7 (Identificador modelo documento), conforme ilustrado na Figura 3.40.

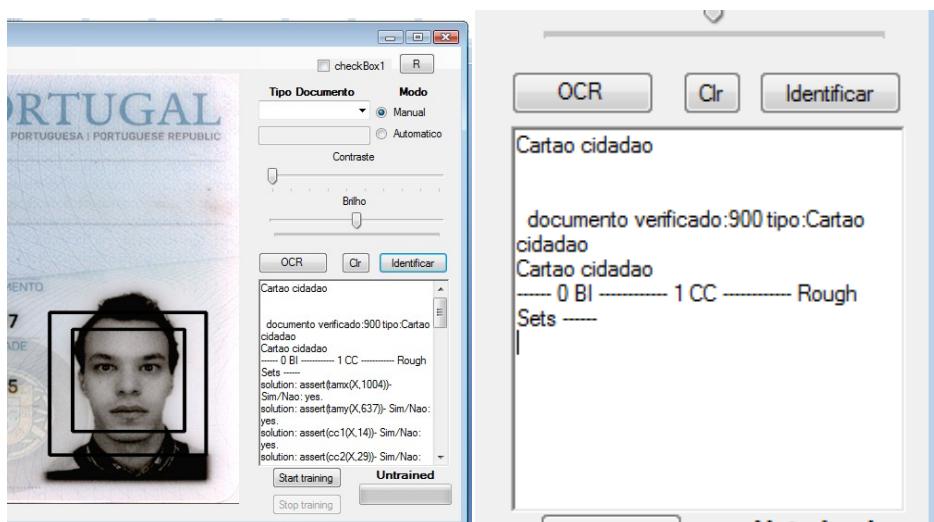


Figura 3.39 - Identificação do tipo de documento

Para se identificar o modelo do documento é necessário comparar com os modelos existentes na base de conhecimento. Um modelo resume as características essenciais de um documento, conforme descrito na Definição 3.4 (Modelo dum documento). Um exemplo dos modelos apresentados por esta definição encontra-se na Figura 3.40, onde se pode constatar parte de um modelo de um tipo de documento.



```

<?xml version="1.0" encoding="utf-8"?>
<ArrayOfDocumento xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:...>
  <Documento>
    <Tipo>BI</Tipo>
    <Dimensao>
      <x>1229</x>
      <y>917</y>
    </Dimensao>
    <nr_campos>2</nr_campos>
    <campos>
      <campo>
        <Nome_campo>Nr BI</Nome_campo>
        <ponto_inicial>
          <x>25</x>
          <y>30</y>
        </ponto_inicial>
        <ponto_final>
          <x>421</x>
          <y>115</y>
        </ponto_final>
      </campo>
      <campo>
        <Nome_campo>Nome</Nome_campo>
        <ponto_inicial>
          <x>25</x>
          <y>110</y>
        </ponto_inicial>
        <ponto_final>
          <x>1000</x>
          <y>216</y>
        </ponto_final>
      </campo>
    </campos>
    <mapa Rows="32" Cols="32" NumberOfChannels="1" CompressionRatio="0">
  </Documento>
</ArrayOfDocumento>

```

Figura 3.40 - Exemplo de modelo de documento

Terminada a identificação do tipo de documento, na imagem do próprio em formato digital, são-lhe identificados os campos úteis a recolher segundo o Modelo desse tipo de documentação. Finalizando este processo, alcança-se o estado descrito na Definição 3.5 (Documento em suporte digital interpretado), definição esta que aplicando o processo OCR (Definição 3.2) é recolhido o texto das zonas úteis e de

forma estruturada, conforme se verifica na Figura 3.35, passando agora para a Definição 3.6 (Leitura interpretado dum documento) que se trata da junção das duas últimas definições enunciadas.

O resultado de todas estas etapas é representado pela Definição 3.8 (Mecanismo de leitura automática dum documento) que fica assim verificada, visto que todas as outras etapas foram também verificadas.

Dado que o sistema LAD seguiu conceptualmente o modelo especificado, e como cada aspecto funcional foi também verificado, resulta na verificação do referido sistema LAD.

3.6.2. Validação

‘A validação dum sistema é realizada para garantir que a especificação do respectivo modelo conceitual captou os aspetos mais importantes de um problema real. As várias abordagens para validar um sistema, incluem a sua utilização em casos críticos, por um lado, e por outro, mediante utilização exaustiva de casos. Pode-se também validar um modelo através de peritos, ou através de simulação’ (Sargent 2005).

Para validação da ferramenta desenvolvida, a sua eficiência, precisão, e bem assim a sua utilidade, foi utilizado um conjunto abrangente de documentos de identificação em formato físico. Os tipos de documentos a identificar foram Bilhetes de Identidade, Cartão de Cidadão e Carta de condução. Utilizaram-se também outros tipos para inferir da precisão e consistência da ferramenta. Os documentos foram todos submetidos ao mesmo processo, que consiste numa digitalização inicial precedido da identificação do tipo de documento e por fim submetidos à leitura estruturada.

Com este método de avaliação obteve-se os resultados, que se podem constatar no gráfico da Figura 3.41.

Este gráfico corresponde ao resultado da identificação dos vários documentos, distribuídos por cores conforme o seu tipo e os vários resultados obtidos, decorrentes da leitura dos documentos, nomeadamente se existe ou não identificação e se existe com ou sem incerteza.

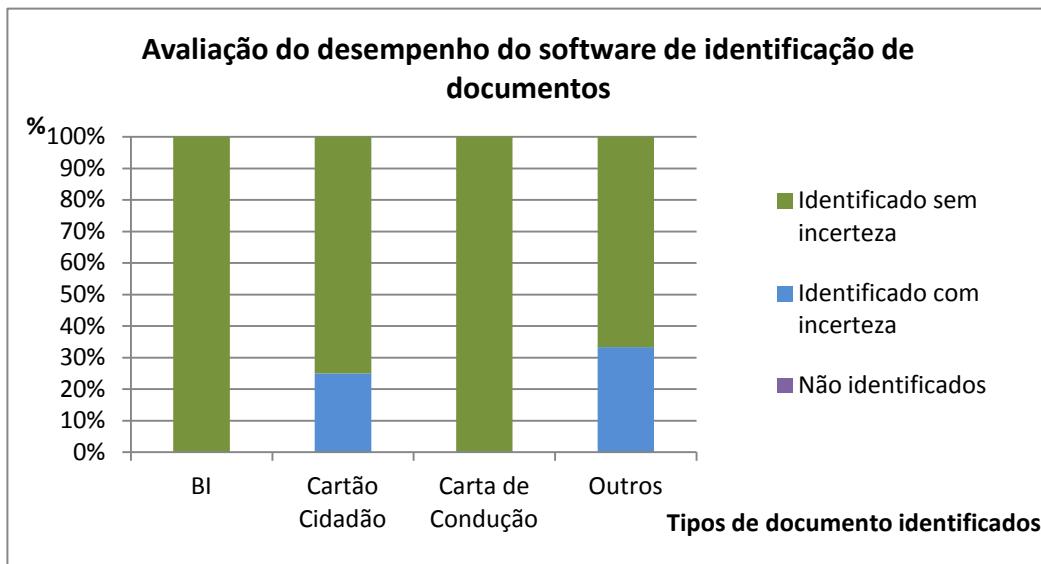


Figura 3.41 - Gáfico de Avaliação de desempenho do software a identificar documentos

Verifica-se uma percentagem elevada de resultados positivos, grande precisão e uma baixa quantidade de falhas.

No gráfico da Figura 3.42 apresentam-se os resultados obtidos aquando da submissão dos documentos à leitura estruturada dos mesmos.

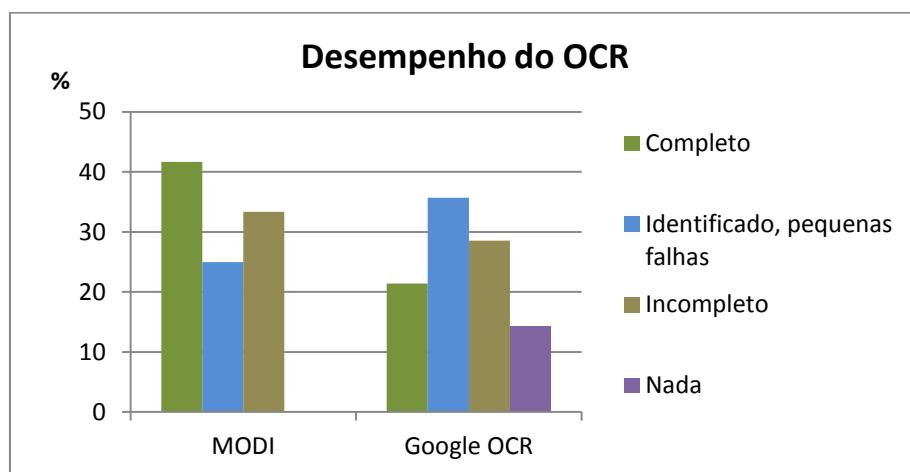


Figura 3.42 - Gráfico de desempenho do bloco OCR conforme a biblioteca aplicada

Constata-se a existência de uma percentagem mais elevada, do que no gráfico anterior, de resultados não correctos e ou equívocos, embora continuando com uma percentagem elevada de reconhecimento correcto de texto, visto que são muito raros os casos em que não se identifique qualquer texto.

Verifica-se assim que a identificação de documentos por parte da ferramenta é precisa e robusta, falhando um pouco aquando da aplicação do OCR. Um aspecto importante, é que o desempenho do sistema varia e pode ter significativas melhorias, dependendo da componente OCR utilizada, conforme também se constata pelos resultados presentes na figura referida.

3.7. Resultados experimentais

Como resultado deste trabalho obtiveram-se vários módulos, os quais foram desenvolvidos tendo em vista o objectivo principal, e levando a uma posterior integração na solução final.

Um dos módulos mais relevantes e desafiantes deste projecto foi o de identificação dos tipos de documentos, tendo sido este baseado em regras de inferência Rough Sets. Tendo-se obtido um desempenho elevado por parte deste módulo e conseguido fazer a identificação de um leque diversificado de documentos com uma grande precisão.

Outro resultado deste projecto foi a implementação de um mecanismo de recolha de informação estrutural de imagens de documentos, permitindo identificar características distintivas dos vários tipos de documentos.

Implementou-se um processo de criação de novos modelos de documento e treino do sistema para posterior identificação dos mesmos, processo este que permite usar uma imagem de um documento para criação de um modelo. Outra vertente deste processo é a selecção automática ou manual dos campos de interesse para esse tipo de documento.

Da integração de todos estes resultados, juntamente com a integração de um mecanismo genérico para digitalização de documentos e um mecanismo OCR genérico obteve-se como resultado final um software modular, que faz a digitalização de documentos, a sua identificação e recolha de informação das zonas de interesse pré-definidas para esse tipo de documento. Fornecendo depois para os SI a informação de forma estruturada. Caso não seja um documento conhecido, este software permite também a criação de um novo modelo de documento, referente a um novo tipo de documento que passa a ser identificável.

4. Conclusões

4.1. Síntese do trabalho efectuado

Neste projeto tentou-se contribuir para a resolução de um problema, que tem suscitado grande interesse e que tem sido investigado e estudado ao longo do tempo, que é o facto de a maioria das ferramentas actualmente existentes para recolha de dados de documentos, não o fazerem de uma forma estruturada.

Os sistemas convencionais de scan com OCR, permitem também a recolha de texto no formato físico, mas sem proporcionar informação num formato estruturado e interpretado.

Como tal, neste projecto propôs-se a produção de uma ferramenta informática que introduzisse mais eficiência e utilidade à recolha de informação precedente de imagens de documentos.

Estabeleceu-se portanto como objetivo deste projecto, a criação de um software em que ao inserirmos um documento de identificação no scanner, o recebe e converte para uma imagem digital, sobre a qual é depois aplicado um processamento prévio e feita a recolha dos dados necessários para a identificação do tipo de documento, precedendo-se à aplicação de OCR.

Dada a parceria com a empresa Softconcept, definiu-se que se daria mais ênfase aos documentos de identificação pessoal. Um exemplo de utilização desta ferramenta seria no momento da realização do check-in num hotel, pois perante um elevado número de clientes tornaria este processo menos trabalhoso e o mais preciso possível.

O sistema também permite reconhecer outro tipo de documentos, como por exemplo, documentos de renda.

Após algumas experiências e pesquisas delineou-se a melhor estrutura a seguir, para assim se cumprir o objetivo pretendido.

A arquitectura definida e implementada mostrou-se bastante efectiva, vindo proporcionar a celeridade de todo o processo e respectivo aumento de eficiência e eficácia.

A rapidez referida advém do elemento mais inovador deste projeto, a utilização de Rough Sets para classificação e correspondente identificação dos tipos de documentos. Esta escolha verificou-se ser, de entre as possibilidades atualmente existentes, a mais acertada e a que deve ser implementada. Neste aspecto, verificou-se no capítulo anterior, aquando da validação do sistema desenvolvido, que os Rough Sets são uma forma bastante simples, rápida e eficiente de se classificar a informação. Sendo que o seu treino não necessita de adaptação aos dados de entrada e a sua execução é rápida e ligeira de recursos computacionais.

Pode-se então concluir que foi cumprido o objectivo desta dissertação, tendo-se obtido um resultado bastante interessante e promissor com a ferramenta desenvolvida. O elemento inovador deste projecto é a utilização de Rough Sets tendo-se verificado ser mais eficiente, quando comparado com outras possibilidades existentes. Temos como exemplo o caso das Redes Neuronais, que requerem um maior processamento prévio dos dados para a sua utilização num treino inicial, que é algo moroso para conseguir ser útil e garantir precisão.

A aplicação do OCR sobre campos específicos, evita perdas de tempo e melhora o seu desempenho, reduzindo também a quantidade de informação recolhida. Os resultados obtidos são bastante interessantes, pois permitem perceber que a verificação do tipo de documento tem uma grande fiabilidade, e que neste sistema o elemento que apresenta pior desempenho e maiores dificuldades é o processo OCR. Foram usados motores opensource, os quais apresentam um desempenho inferior quando comparados com opções licenciadas, tal pode ser rectificado visto o projecto ter sido pensado e executado numa lógica de blocos podendo este ser substituído por um outro motor OCR.

4.2. Trabalho Futuro

Os resultados anteriormente apresentados, demonstram que os passos futuros a ter em conta passam pela melhoria ou substituição do mecanismo de OCR por um de melhor qualidade. Normalmente, isto faz-se recorrendo à utilização de uma biblioteca comercial. Dependendo do tipo de documento, texto e imagens contidas, podem-se especificar quais as técnicas de tratamento de imagem que irão ser utilizadas em cada campo, dentro de um conjunto de técnicas pré-estabelecido.

Outro elemento que pode ser tido como área de desenvolvimento futuro, será aumentar o número de características utilizadas na identificação do tipo de documento, ou então aumentar a matriz utilizada na recolha de dados.

Para recolha de características de uma imagem de um documento, utiliza-se neste momento uma matriz 3x3. Caso se tenha capacidade de processamento poderá utilizar-se uma matriz maior, que irá permitir uma maior assertividade e detalhe na identificação, evitando assim casos de falsa identificação de documentos ou a não validação da identificação.

O aumento desta matriz será útil caso se pretenda aplicar o sistema em casos reais, em que será utilizada uma maior diversidade de documentos, o que se apresenta como um acréscimo de dificuldade para o software na distinção do tipo de documentos. Se limitarmos o leque de documentos que se pode identificar, como foi feito neste projeto, a sua eficácia aumenta.

Assim sendo, este documento termina com a esperança que num futuro próximo se consiga construir uma ferramenta totalmente autónoma, similar à desenvolvida neste projeto, que processe um documento de identificação, mas não só, que recolha os dados pertinentes e de interesse de forma estruturada, e quando for necessário permita a criação de novos modelos de documentos sem intervenção externa, ou que faça um aconselhamento inteligente, facilitando a vida do operador.

Bibliografia

- Abdulkader, A.C.M.R., 2009. Low cost correction of OCR errors using learning in a multi-engine environment. *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference*, pp.576 – 580 . Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5277588 [Accessed July 21, 2012].
- B.H.Thacker et al., 2004. *Concepts of Model Verification and Validation*, Los Alamos, NM. Available at: <http://www.osti.gov/scitech/biblio/835920> [Accessed September 25, 2013].
- Breuel, T.M., 2008. The OCropus open source OCR system. *Proceedings IS&T/SPIE 20th Annual Symposium*. Available at: <http://144.206.159.178/FT/CONF/16408773/16408787.pdf> [Accessed July 21, 2012].
- Chakraborty, R., 2010. Fundamentals of Neural Networks Soft Computing
Fundamentals of Neural Networks Soft Computing.
- Deerwester, S., Dumais, S. & Landauer, T., 1990. Indexing by latent semantic analysis. *JAS/S*. Available at: http://www.cob.unt.edu/itds/faculty/evangelopoulos/dsci5910/LSA_Deerwester1990.pdf [Accessed September 18, 2013].
- Egozi, O., Markovitch, S. & Gabrilovich, E., 2011. Concept-Based Information Retrieval Using Explicit Semantic Analysis. *ACM Transactions on Information Systems*, 29(2), pp.1–34. Available at: <http://dl.acm.org/citation.cfm?id=1961209.1961211> [Accessed September 18, 2013].
- Gonzalez, R.C. & Woods, R.E., 2002. *Digital Image Processing* Second Edi., Prentice-Hall. Available at: http://scholar.google.com/scholar?cluster=4109123112636618287&hl=en&num=20&as_sdt=2005&sciodt=0,5#0 [Accessed August 7, 2013].
- Haikin, S., 1998. Neural Networks: A Comprehensive Foundation. Available at: <http://www.citeulike.org/group/296/article/431637> [Accessed February 10, 2012].
- Haykin, S., 1994. Neural Networks: A Comprehensive Foundation. Available at: <http://dl.acm.org/citation.cfm?id=541500> [Accessed September 4, 2013].

- HAYKIN, S.S., 2001. *Redes Neurais - 2ed.*, Available at:
<http://www.google.pt/books?hl=pt-PT&lr=&id=IBp0X5qfyjUC&pgis=1> [Accessed September 4, 2013].
- Hirayama, J. et al., 2011. Development of Template-Free Form Recognition System. In *2011 International Conference on Document Analysis and Recognition*. IEEE, pp. 237–241. Available at:
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6065311' escapeXml='false' /> [Accessed December 12, 2012].
- Hvidsten, T.R., 2010. A tutorial-based guide to the ROSETTA system : A Rough Set Toolkit for Analysis of Data.
- Kauniskangas, H., 1999. Document Image Retrieval With Improvements In Database Quality. Available at:
<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.41.2899> [Accessed September 5, 2013].
- Kohonen, T., 2001. Self-organizing maps. *Springer*, 30.
- Kohonen, T., 1990. The Self-Organizing Map. *Proceedings of IEEE*, 78(9), pp.1464–1480. Available at: [http://www.eicstes.org/EICSTES_PDF/PAPERS/The%20Self-Organizing%20Map%20\(Kohonen\).pdf](http://www.eicstes.org/EICSTES_PDF/PAPERS/The%20Self-Organizing%20Map%20(Kohonen).pdf) [Accessed September 4, 2013].
- Lee, J.-H. et al., 2009. Automatic generic document summarization based on non-negative matrix factorization. *Information Processing & Management*, 45(1), pp.20–34. Available at:
<http://www.sciencedirect.com/science/article/pii/S030645730800068X> [Accessed September 18, 2013].
- Lienhart, R. & Maydt, J., 2002. An extended set of Haar-like features for rapid object detection. In *Proceedings. International Conference on Image Processing*. IEEE, pp. I–900–I–903. Available at:
<http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=1038171&contentType=Conference+Publications> [Accessed August 20, 2012].
- Ministros, A. para a M.A.I.-P. do C. de, 2009. Cartão do Cidadão. Available at:
http://www.cartaodecidadao.pt/index.php?option=com_content&task=view&id=19&Itemid=29&lang=pt.html [Accessed March 10, 2013].
- Mori, S., Suen, C.Y. & Yamamoto, K., 1992. Historical review of OCR research and development. *Proceedings of the IEEE*, 80(7), pp.1029–1058. Available at:
<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=156468>.
- Øhrn, A., 2000. *Discernibility and rough sets in medicine: tools and applications*. Norwegian University of Science and Technology N-7491 Trondheim, Norway. Available at: <http://ntnu.diva-portal.org/smash/record.jsf?pid=diva2:125243> [Accessed February 21, 2013].
- Otsu, N., 1975. A threshold selection method from gray-level histograms. *Automatica*. Available at: <http://www.tecgraf.puc-rio.br/~mgattass/cg/trblImg/Otsu.pdf> [Accessed March 21, 2013].

- Pawlak, Z., 1997. Rough set approach to knowledge-based decision support. *European Journal of Operational Research*, 99(1), pp.48–57. Available at: [http://dx.doi.org/10.1016/S0377-2217\(96\)00382-7](http://dx.doi.org/10.1016/S0377-2217(96)00382-7) [Accessed February 10, 2013].
- Pawlak, Z., 1982. Rough sets. *International Journal of Computer and Information Sciences*, 11(5), pp.341–356. Available at: <http://www.springerlink.com/content/r5556398717921x5/>.
- Pawlak, Z., 1995. Rough sets (abstract). In *Proceedings of the 1995 ACM 23rd annual conference on Computer science - CSC '95*. New York, New York, USA: ACM Press, pp. 262–264. Available at: <http://dl.acm.org/citation.cfm?id=259526.277421> [Accessed September 18, 2012].
- Pawlak, Z., 2002. Rough sets and intelligent data analysis. *Information Sciences*, 147(1-4), pp.1–12. Available at: [http://dx.doi.org/10.1016/S0020-0255\(02\)00197-4](http://dx.doi.org/10.1016/S0020-0255(02)00197-4) [Accessed September 18, 2012].
- Pawlak, Z. & Skowron, A., 2007. Rudiments of rough sets. *Information Sciences*, 177(1), pp.3–27. Available at: <http://dx.doi.org/10.1016/j.ins.2006.06.003> [Accessed February 10, 2013].
- Pratt, W.K., 2007. *Digital Image Processing* Fourth. I. PixelSoft, ed., Los Altos, California: WILEY-INTERSCIENCE A John Wiley & Sons, Inc., Publication.
- Radinsky, K. et al., 2011. A word at a time. In *Proceedings of the 20th international conference on World wide web - WWW '11*. New York, New York, USA: ACM Press, p. 337. Available at: <http://dl.acm.org/citation.cfm?id=1963405.1963455> [Accessed September 18, 2013].
- Radzikowska, A.M. & Kerre, E.E., 2002. A comparative study of fuzzy rough sets. *Fuzzy Sets and Systems*, 126(2), pp.137–155. Available at: [http://dx.doi.org/10.1016/S0165-0114\(01\)00032-X](http://dx.doi.org/10.1016/S0165-0114(01)00032-X) [Accessed September 18, 2012].
- Rankl, W. & Effing, W., 2010. *Smart Card Handbook* 4th ed. J. Wiley & Sons, ed., Available at: <http://www.google.pt/books?hl=pt-PT&lr=&id=C55-4kVUQ14C&pgis=1> [Accessed August 19, 2013].
- Ripley, B., 2008. *Pattern recognition and neural networks*, Cambridge, United Kingdom: Press Syndicate of THE UNIVERSITY OF CAMBRIDGE.
- Sargent, R.G., 2005. Verification and validation of simulation models. , pp.130–143. Available at: <http://dl.acm.org/citation.cfm?id=1162708.1162736> [Accessed September 25, 2013].
- Smith, R., 2007. An overview of the Tesseract OCR engine. *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference*, 2, pp.629 – 633. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4376991 [Accessed July 21, 2012].
- Smith, R., Antonova, D. & Lee, D., 2009. Adapting the Tesseract open source OCR engine for multilingual OCR. ... *Multilingual OCR* ..., (Cc). Available at: <http://tesseract-ocr.googlecode.com/svn->

history/r484/trunk/doc/MOCRadaptingtesseract2.pdf [Accessed February 21, 2013].

Softconcept, Softconcept. Available at: <http://www.softconcept.pt/>.

Stollnitz, E.J., DeRose, T.D. & Salesin, D.H., 1995. Wavelets for computer graphics: a primer part. *IEEE Computer Graphics and Applications*, 15(May), pp.1–8. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=376616 [Accessed March 5, 2013].

Struzik, Z.R. et al., 1999. The Haar Wavelet Transform in the Time Series Similarity Paradigm. In J. M. Żytkow & J. Rauch, eds. *Principles of Data Mining and Knowledge Discovery*. Berlin, Heidelberg: Springer Berlin Heidelberg. Available at: <http://www.springerlink.com/content/pypmgcc2wut57nbb/> [Accessed September 19, 2012].

Viola, P. & Jones, M., 2004. Rapid object detection using a boosted cascade of simple features. ... *Vision and Pattern Recognition, 2001. CVPR ...*, 1, pp.I-511–I-518. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=990517 [Accessed July 25, 2012].

Way, O.M., Jones, M.J. & Viola, P., 2004. Robust Real-Time Face Detection. *International Journal of Computer Vision*, 57(2), pp.137–154. Available at: <http://link.springer.com/10.1023/B:VISI.0000013087.49260.fb> [Accessed February 10, 2013].

Anexos

Regras								
CC7(22) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC7(45) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC7(60) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC7(76) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC7(87) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC7(100) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC7(110) => Tip0(0)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC7(117) => Tip0(1)	1	1	1.0	0.076923	0.5	1.0	1	1
CC7(124) => Tip0(1)	1	1	1.0	0.076923	0.5	1.0	1	1
CC7(137) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC7(146) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC7(158) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC7(164) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC9(19) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC9(34) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC9(50) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC9(74) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC9(85) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC9(106) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC9(110) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC9(114) => Tip0(1)	1	1	1.0	0.076923	0.5	1.0	1	1
CC9(118) => Tip0(1)	1	1	1.0	0.076923	0.5	1.0	1	1
CC9(122) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC9(126) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC9(130) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC9(134) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
F_Y(1) => Tip0(2)	6	6	1.0	0.461538	1.0	1.0	1	1
F_Y(0) => Tip0(0)	5	5	1.0	0.384615	1.0	1.0	1	1
F_Y(2) => Tip0(1)	2	2	1.0	0.153846	1.0	1.0	1	1
CC1(78) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC1(120) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC1(154) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC1(196) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC1(229) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC1(272) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC1(280) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC1(294) => Tip0(1)	1	1	1.0	0.076923	0.5	1.0	1	1
CC1(308) => Tip0(1)	1	1	1.0	0.076923	0.5	1.0	1	1
CC1(321) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	1	1

CC1(330) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC1(352) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC1(361) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC6(34) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC6(44) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC6(47) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC6(50) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC6(52) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC6(60) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC6(79) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC6(82) => Tipo(1)	1	1	1.0	0.076923	0.5	1.0	1	1
CC6(85) => Tipo(1)	1	1	1.0	0.076923	0.5	1.0	1	1
CC6(103) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC6(123) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC6(140) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC6(155) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC5(120) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC5(145) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC5(170) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC5(196) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC5(218) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC5(255) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC5(302) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC5(320) => Tipo(1)	1	1	1.0	0.076923	0.5	1.0	1	1
CC5(338) => Tipo(1)	1	1	1.0	0.076923	0.5	1.0	1	1
CC5(380) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC5(425) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC5(468) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC5(503) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
Tamy(641) => Tipo(2)	2	2	1.0	0.153846	0.333333	1.0	1	1
Tamy(974) => Tipo(0)	2	2	1.0	0.153846	0.4	1.0	1	1
Tamy(845) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
Tamy(844) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
Tamy(941) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
Tamy(637) => Tipo(1)	2	2	1.0	0.153846	1.0	1.0	1	1
Tamy(638) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
Tamy(626) => Tipo(2)	2	2	1.0	0.153846	0.333333	1.0	1	1
Tamy(634) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC4(41) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC4(68) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC4(102) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC4(120) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC4(154) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC4(182) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC4(189) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC4(199) => Tipo(1)	1	1	1.0	0.076923	0.5	1.0	1	1
CC4(209) => Tipo(1)	1	1	1.0	0.076923	0.5	1.0	1	1
CC4(221) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC4(228) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC4(239) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC4(247) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC2(144) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC2(203) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC2(238) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC2(296) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC2(332) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC2(390) => Tipo(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC2(441) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC2(470) => Tipo(1)	1	1	1.0	0.076923	0.5	1.0	1	1
CC2(499) => Tipo(1)	1	1	1.0	0.076923	0.5	1.0	1	1
CC2(552) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC2(604) => Tipo(2)	1	1	1.0	0.076923	0.166667	1.0	1	1

CC2(552) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC2(604) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC2(659) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC2(710) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC8(81) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC8(91) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC8(98) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC8(106) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC8(111) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC8(118) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	1	1
CC8(168) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC8(193) => Tip0(1)	1	1	1.0	0.076923	0.5	1.0	1	1
CC8(218) => Tip0(1)	1	1	1.0	0.076923	0.5	1.0	1	1
CC8(253) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC8(303) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC8(338) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
CC8(379) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	1	1
Face(1) AND F_X(0) => Tip0(2)	6	6	1.0	0.461538	1.0	1.0	2	1
Face(0) AND F_X(0) => Tip0(0)	5	5	1.0	0.384615	1.0	1.0	2	1
Face(1) AND F_X(2) => Tip0(1)	2	2	1.0	0.153846	1.0	1.0	2	1
Tamx(1004) AND F_X(0) => Tip0(2)	4	4	1.0	0.307692	0.666667	1.0	2	1
Tamx(1232) AND F_X(0) => Tip0(0)	2	2	1.0	0.153846	0.4	1.0	2	1
Tamx(1228) AND F_X(0) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	2	1
Tamx(1224) AND F_X(0) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	2	1
Tamx(1216) AND F_X(0) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	2	1
Tamx(1008) AND F_X(0) => Tip0(2)	2	2	1.0	0.153846	0.333333	1.0	2	1
Tamx(1004) AND F_X(2) => Tip0(1)	2	2	1.0	0.153846	1.0	1.0	2	1
CC3(71) AND F_X(0) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	2	1
CC3(92) AND F_X(0) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	2	1
CC3(105) AND F_X(0) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	2	1
CC3(128) AND F_X(0) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	2	1
CC3(140) AND F_X(0) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	2	1
CC3(163) AND F_X(0) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	2	1
CC3(180) AND F_X(0) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	2	1
CC3(180) AND F_X(2) => Tip0(1)	2	2	1.0	0.153846	1.0	1.0	2	1
CC3(191) AND F_X(0) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	2	1
CC3(210) AND F_X(0) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	2	1
CC3(216) AND F_X(0) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	2	1
CC3(234) AND F_X(0) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	2	1
Tamx(1004) AND CC3(71) AND Face(1) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	3	1
Tamx(1232) AND CC3(92) AND Face(0) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	3	1
Tamx(1232) AND CC3(105) AND Face(0) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	3	1
Tamx(1228) AND CC3(128) AND Face(0) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	3	1
Tamx(1224) AND CC3(140) AND Face(0) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	3	1
Tamx(1216) AND CC3(163) AND Face(0) => Tip0(0)	1	1	1.0	0.076923	0.2	1.0	3	1
Tamx(1008) AND CC3(180) AND Face(1) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	3	1
Tamx(1004) AND CC3(180) AND Face(1) => Tip0(1)	2	2	1.0	0.153846	1.0	1.0	3	1
Tamx(1004) AND CC3(191) AND Face(1) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	3	1
Tamx(1004) AND CC3(210) AND Face(1) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	3	1
Tamx(1008) AND CC3(216) AND Face(1) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	3	1
Tamx(1004) AND CC3(234) AND Face(1) => Tip0(2)	1	1	1.0	0.076923	0.166667	1.0	3	1

Tabela 0.1 - Tabela de regras usada no software para classificação dos documentos