



Ricardo Almeida Venieris

**Uma Arquitetura de Software para Apoio
ao Desenvolvimento de Sistemas de
Diagnóstico Médicos por Imagem**

Dissertação de Mestrado

Dissertação apresentada ao Programa de Pós-graduação em Informática da PUC-Rio como requisito parcial para obtenção do grau de Mestre em Informática.

Orientador: Prof. Carlos José Pereira de Lucena

Rio de Janeiro

Fevereiro de 2018



Ricardo Almeida Venieris

**Uma Arquitetura de Software para Apoio ao Desenvolvimento de
Sistemas de Diagnóstico Médicos por Imagem**

Dissertação apresentada como requisito parcial
para obtenção do grau de Mestre pelo Programa
de Pós-graduação em Informática da PUC-Rio.
Aprovada pela Comissão Examinadora abaixo
assinada.

Prof. Carlos José Pereira de Lucena

Orientador

Departamento de Informática – PUC-Rio

Prof. Hélio Côrtes Vieira Lopes

Departamento de Informática – PUC-Rio

Prof. Marx Leles Viana

Pesquisador Autônomo

Prof. Márcio da Silveira Carvalho

Coordenador Setorial do Centro

Técnico Científico – PUC-Rio

Rio de Janeiro, 07 de fevereiro de 2018

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Ricardo Almeida Venieris

Graduou-se em Bacharel em Sistemas de Informação pela Pontifícia Universidade Católica do Rio de Janeiro em 2013. Atualmente atua na área de Engenharia de Software no Laboratório de Engenharia de Software (LES) da PUC-Rio.

Ficha Catalográfica

Venieris, Ricardo Almeida.

Uma Arquitetura de Software para Apoio ao Desenvolvimento de Sistemas de Diagnóstico Médicos por Imagem / Ricardo Almeida Venieris; orientador: Carlos José Pereira de Lucena – Rio de Janeiro PUC, Departamento de Informática, 2018.

v., 99 f.: il. ; 29,7 cm

1. Dissertação (mestrado) – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática.

Inclui referências bibliográficas.

1. Informática – Teses. 2. Engenharia de Software 3. Arquitetura de Software 4. Arquitetura de Software para Apoio ao Desenvolvimento de Sistemas de Diagnóstico. I. Lucena, Carlos José Pereira de. II. Pontifícia Universidade Católica do Rio de Janeiro. III. Departamento de Informática IV. Título.

CDD: 004

**À minha filha, motor de minha vontade e persistência,
a que me fez encontrar as perguntas certas.**

Agradecimentos

Ao meu pai Panaiotis, por me ensinar com os melhores exemplos a importância da perseverança e força de trabalho.

À minha mãe Nilza, por me ensinar através de sua dedicação e ternura o que é amor, união, força de equipe e gratidão.

À minha filha Beatriz, por só me dar orgulho, por ser o motor que impulsiona minha vida, tornando-me um homem melhor.

À minha esposa Silvia, pelo apoio, compreensão e por tornar minha vida mais feliz nestes últimos anos.

Ao meu irmão Marcelo, pelo companheirismo e suporte em todos os momentos de minha vida

À minha irmã Dra. Monica, minha companheira e inspiração, sem ela este trabalho não teria acontecido.

Ao meu orientador Prof. Carlos Lucena, presente em minha vida acadêmica desde 1993, um farol no mar da informática.

À todos do Laboratório de Engenharia de Software e da Apple Developer Academy PUC-Rio, especialmente Gustavo de Carvalho, Andrew Costa e Hendi Lemos, pela oportunidade oferecida, apoio e confiança. Minha graduação não seria possível sem o suporte do Laboratório.

À todos os amigos que sempre estiveram ao meu lado, na alegria e na tristeza.

E a Deus, por oferecer-me a chance de viver minha “terceira vida nesta vida” de maneira plena e emocionante.

Resumo

Venieris, Ricardo Almeida; Lucena, Carlos José Pereira de (Orientador). **Uma Arquitetura de Software para Apoio ao Desenvolvimento de Sistemas de Diagnóstico Médicos por Imagem**. Rio de Janeiro, 2018. 99p. Dissertação de Mestrado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

O apoio diagnóstico de exames médicos por imagem utilizando técnicas de Inteligência Artificial tem sido amplamente discutido e pesquisado academicamente. Diversas técnicas computacionais para segmentação e classificação de tais imagens são continuamente criadas, testadas e aperfeiçoadas. Destes estudos emergem sistemas com alto grau de especialização que se utilizam de técnicas de visão computacional e aprendizagem de máquina para segmentar e classificar imagens de exames utilizando conhecimento adquirido através de grandes coleções de exames devidamente laudados. No domínio médico há ainda a dificuldade de se conseguir bases de dados qualificada para realização da extração de conhecimento pelos sistemas de aprendizagem de máquina. Neste trabalho propomos a construção de uma arquitetura de software que suporte o desenvolvimento de sistemas de apoio diagnóstico que possibilite: (i) a utilização em múltiplos tipos exames, (ii) que consiga segmentar e classificar, (iii) utilizando não só de estratégias padrão de aprendizado de máquina como, (iv) o conhecimento do domínio médico disponível. A motivação é facilitar a tarefa de geração de classificadores que possibilite, além de buscar marcadores patológicos específicos, ser aplicado em objetivos diversos da atividade médica, como o diagnóstico pontual, triagem e priorização do atendimento.

Palavras-chave

Inteligência Artificial; Aprendizagem de Máquina; Visão Computacional; Arquitetura de Software; Apoio a triagem diagnóstica; *Entity-component-system*.

Abstract

Venieris, Ricardo Almeida; Lucena, Carlos José Pereira de (Advisor). **A Software Architecture to Support Development of Medical Imaging Diagnostic Systems**. Rio de Janeiro, 2018. 99p. Dissertação de Mestrado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

The image medical exam diagnostic support using Artificial Intelligence techniques has been extensively discussed and academically researched. Several computational techniques for segmentation and classification of such images are continuously created, tested and improved. From these studies, highly specialized systems that use computational vision and machine learning techniques to segment and classify exam images using knowledge acquired through large collections of lauded exams. In the medical domain, there is still the difficulty of obtaining qualified databases to support the extraction of knowledge by machine learning systems. In this work we propose a software architecture construction that supports diagnostic support systems development that allows: (i) use of multiple exam types, (ii) supporting segmentation and classification, (iii) using not only machine learning techniques as, (iv) knowledge of the available medical domain. The motivation is to facilitate the generation of classifiers task that, besides searching for specific pathological markers, can be applied to different medical activity objectives, such as punctual diagnosis, triage and prioritization of care.

Keywords

Artificial Intelligence; Machine Learning; Computer Vision; Software Architecture; Diagnostic Screening Support; Entity-component-system.

Sumário

1	Introdução	11
1.1	Motivação	12
1.2	O Problema	16
1.3	Limitações nos Trabalhos Referenciados	17
1.4	Solução Proposta	18
1.4.1	Abordagem e estratégia de Aprendizado de Máquina	18
1.5	Coleta de Informações para Estabelecimento da Base de Pesquisa	19
1.6	Contribuições Esperadas	23
1.7	Organização da Dissertação	23
2	Fundamentação Teórica	24
2.1	KDD	24
2.2	Aprendizagem de Máquina	27
2.3	Sistemas de Visão Computacional	32
2.3.1	Aquisição de Imagens	33
2.3.2	Pré-processamento	34
2.3.3	Segmentação	34
2.3.4	Extração de Atributos	34
2.3.5	Reconhecimento e Interpretação	35
2.4	Detecção de Anomalia	35
2.5	Entity-Component-System	36
2.6	Framework	42
2.7	Arquitetura de Software	43
3	Trabalhos Relacionados	45
3.1	Sistema de Visão Computacional para Detecção e Quantificação de Enfisema Pulmonar, (FÉLIX, 2007)	46
3.2	Desenvolvimento de um Sistema de Auxílio ao Diagnóstico de Pneumonia na Infância Utilizando Visão Computacional, (MACEDO et al., 2012)	50
3.3	Semantic Entity-Component State Management Techniques to Enhance Software Quality for Multimodal VR-Systems, (FISCHBACH; WIEBUSCH; LATOSCHIK, 2017)	52

4	Solução Proposta	55
4.1	O Processo	56
4.1.1	Novo Exame	57
4.1.2	Aquisição de Artefato	57
4.1.3	Pré-processamento	58
4.1.4	Segmentação	58
4.1.5	Extração de Atributos Orientados por Domínio	61
4.1.6	Classificação	63
4.1.7	Interpretação	65
4.2	Diagrama de Classes do Modelo	66
4.3	Implementação Usando ECS	69
4.3.1	Implementação Exemplo: Avaliação do Risco de Eventos Cardiovasculares Através de Ultrassonografia da Carótida com EIMC	71
4.3.2	Implementação Exemplo: Classificador de Fundoscopias com Presença de Exsudatos Duros	74
4.3.3	Implementação Exemplo: Classificador de Radiografias de Tórax com Presença de Pneumonias	76
5	Conclusão e Trabalhos Futuros	78
5.1	Contribuições	80
5.2	Trabalhos Futuros	81
6	Referências bibliográficas	83
7	Apêndices	95
7.1	Apêndice A – Situação da Cobertura médica oftalmológica no Brasil	95
7.2	Apêndice B – Considerações complementares sobre a abordagem, o ato médico e seus fundamentos legais	97

Lista de figuras

Figura 1 - Total de exames de ressonância magnética, por 1.000 habitantes/ano, para os países membros e parceiros da OCDE. (OECD, 2015b)..	13
Figura 2 - Total de exames de tomografia computadorizada, para cada 1.000 habitantes/ano. (OECD, 2015b).....	13
Figura 3 - Exemplo de segmentação de imagem 3D (RSNA, 2017).....	16
Figura 4 - Exemplos de imagens de fundo de olho com exsudatos duros (A) saudáveis (B) e (C) com outra doença não classificada	19
Figura 5 - Fases do processo KDD (FAYYAD et al., 1996).....	25
Figura 6 - Exemplo de transformação de valores contínuos para categóricos...	26
Figura 7 - Sistema típico de visão computacional (FILHO; NETO, 1999).....	33
Figura 8 - Exemplo de classificação ideal de instâncias normais, anomalias e ruídos.	36
Figura 9 - Tentativa de modelar o reino animal utilizando hierarquia.....	39
Figura 10 - Modelagem do reino animal utilizando ECS.....	40
Figura 11 - Implementação base de um ECS.....	41
Figura 12 - Imagem de exame pulmonar (A), seu histograma (B) e limiarização (C) (FÉLIX, 2007).....	47
Figura 13 - Perfil de inundação das bacias de captação com represa (A) e exemplo de segmentação (B) através de transformada de <i>watershed</i> (FÉLIX, 2007).....	47
Figura 14 - Estruturas do aparelho pulmonar (FÉLIX, 2007).....	48
Figura 15 - Avaliação da segmentação para cada algoritmo estudado (FÉLIX, 2007).....	49
Figura 16 - Exemplos de interfaces multimodais (MACEDO et al., 2012).....	52
Figura 17 - Fluxo de processo para interpretação de exame.....	56
Figura 18 - Exemplo de métodos de segmentação de uma imagem de ressonância magnética da coluna lombar	60
Figura 19 - Ultrassonografia da carótida com EIMC	62
Figura 20 - Pacotes e classes da camada de modelo do sistema proposto.....	66
Figura 21 - Radiografia de tórax normal em PA apontando as principais estruturas anatômicas identificáveis	68

Figura 22 - ECS com identificação de <i>Hot-spots</i> e <i>Frozen-spots</i>	69
Figura 23 - Frozen-spots da implementação do processo proposto	70
Figura 24 - Objetos instanciados do <i>Entity-component-system</i> proposto para classificação de Ultrassonografia de Carótida com EIMC	72
Figura 25 - Objetos instanciados do <i>Entity-component-system</i> proposto para classificação de imagens possuidoras de exsudatos duros	75
Figura 26 - Relação oftalmologista/população por estado Brasileiro. (CBO, 2014).....	96
Figura 27 - Formação da ISO 17025	97

1 Introdução

O apoio diagnóstico de exames médicos por imagem utilizando técnicas de IA (Inteligência Artificial) tem sido amplamente discutido e pesquisado academicamente. São alguns exemplos os trabalhos apresentados por Abramoff & Niemeijer (2008), Cunha e outros (2016) e Gardezi Eltoukhy & Faye (2017). Diversas técnicas de computação para segmentação e classificação de imagens de exames médicos são continuamente criadas, testadas e aperfeiçoadas, afirmando a viabilidade de certos algoritmos para a realização das tarefas de classificação e segmentação de imagens, assim como analisado o esforço computacional destes algoritmos (BOTTOU, 2010).

Os trabalhos de desenvolvimento deste tipo de tecnologia não costumam dialogar com a academia da área médica, focando as pesquisas nas áreas da informática de engenharia de software e inteligência artificial. Pesquisas orientadas são extremamente necessárias para o desenvolvimento da tecnologia e ampliação de suas possibilidades, os sistemas derivados destes trabalhos são comumente desenvolvidos *ad hoc* e, justamente por não ser ponto focal da pesquisa de Engenharia de Software, a resposta da área médica a tais sistemas foi pouco mencionada nos trabalhos referenciados por esta dissertação.

A pretensão deste trabalho é evoluir na área de engenharia de software, para apoiar a construção de sistemas de inteligência artificial aplicados ao domínio da medicina, que possibilitem e facilitem o reuso e grupamento do conhecimento nos diversos trabalhos que abordam este tema, materializado em forma de modelos de classificação, segmentação, algoritmos e atributos de interesse do domínio médico.

O entendimento dos desafios práticos, dos problemas atuais e reais, a amplitude e diversidade da área médica é o principal orientador para o desenvolvimento deste trabalho.

1.1

Motivação

Apesar dos grandes avanços nas pesquisas desenvolvidas utilizando IA para realização da tarefa de identificação de doenças em exames médicos por imagem, não foram identificadas, nos trabalhos referenciados por essa dissertação (discriminados na seção 1.5 deste documento), soluções que possam ser facilmente utilizados para diversos tipos de exames, usando diversas técnicas de segmentação e classificação das imagens. Os sistemas avaliados possuem escopo fechado na resolução de um único problema, uma doença ou categoria de doenças para um único tipo de exame, identificando as características únicas de uma específica doença.

Em contrapartida, novas tecnologias estão nos possibilitando executar exames por imagem por custos cada vez mais baixos, com qualidade cada vez maior (DATASUS, 2015). Este fato está permitindo a execução de campanhas de coleta de exames em grandes massas populacionais para diagnóstico preventivo (FIOCRUZ, 2017). Outra informação relevante para entendimento da motivação e do cenário de aplicabilidade é o fato de o Brasil estar entre os países que mais produz imagens de exames per capita (ANS, 2016).

Na Figura 1 é apresentado o número total de exames de ressonância magnética, realizados em hospitais e na atenção ambulatorial, por 1.000 habitantes, para os países membros e parceiros da Organização para a Cooperação e Desenvolvimento Econômico (OCDE). É fato que o número de exames de ressonância magnética executados pelas organizações de saúde complementar (saúde privada) está em primeiro lugar. Na Figura 2 temos os números equivalentes para exames de tomografia computadorizada, onde o Brasil está acima da média da OCDE. (Fonte: OECD (2015b), SIP/ANS/MS - 03/2016 e SIB/ANS/MS - 03/2016)

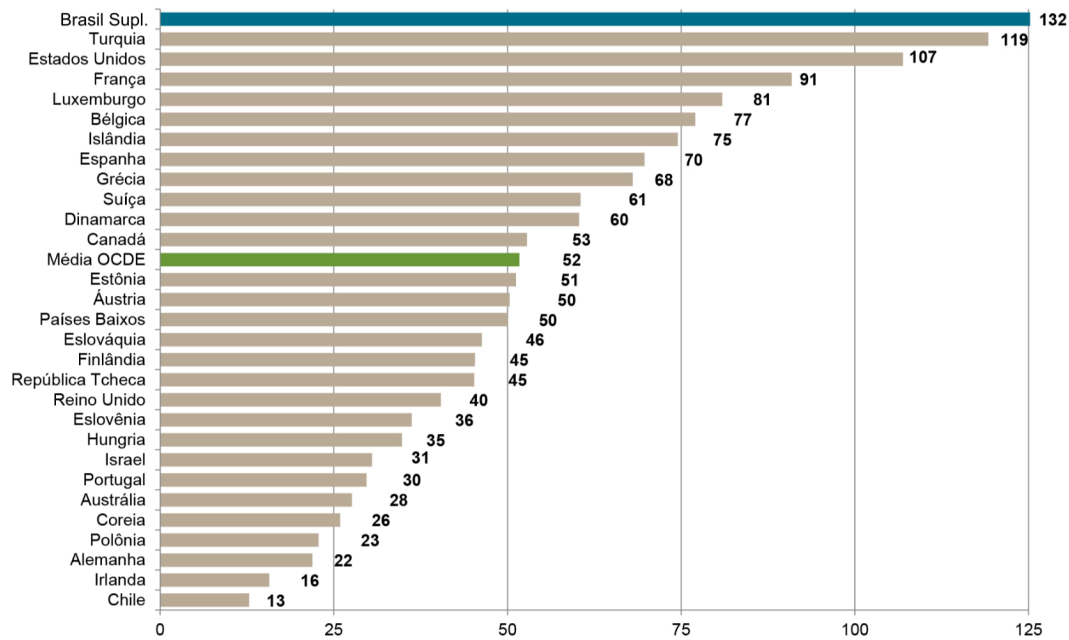


Figura 1 - Total de exames de ressonância magnética, por 1.000 habitantes/ano, para os países membros e parceiros da OCDE. (OECD, 2015b)

PUC-Rio - Certificação Digital Nº 1521389/CA

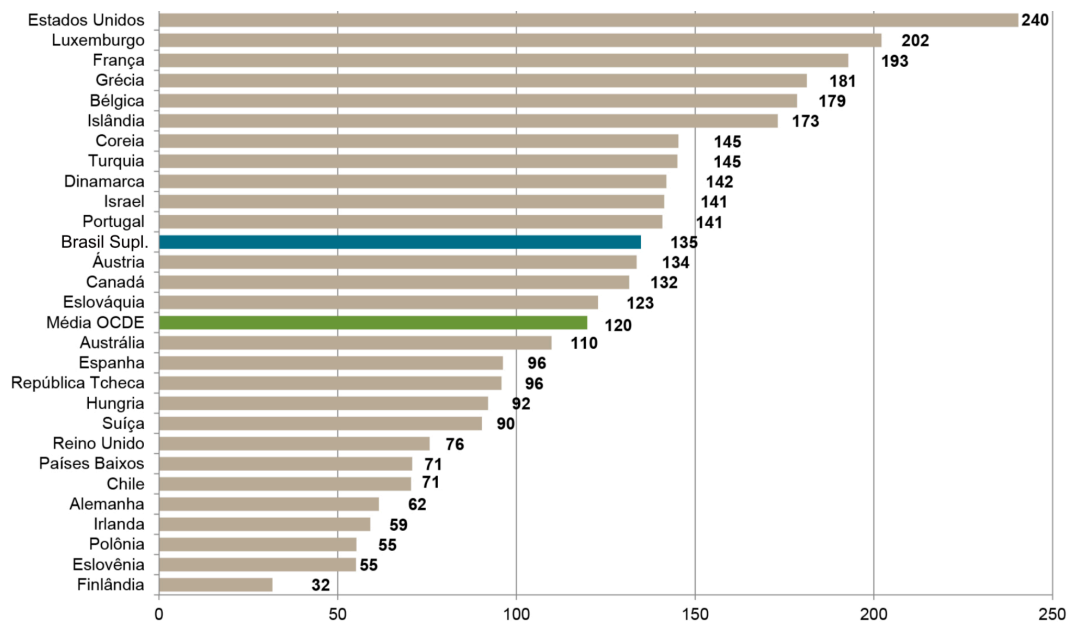


Figura 2 - Total de exames de tomografia computadorizada, para cada 1.000 habitantes/ano. (OECD, 2015b)

O tema “*esforço de desenvolvimento de sistemas de apoio diagnóstico X benefícios práticos aplicáveis*” toma grande relevância ao trabalharmos no campo de diagnósticos de doenças dado que, segundo a Classificação Internacional de Doenças – 10ª edição (CID-10) da Organização Mundial de Saúde (OMS) temos 14.233 doenças catalogadas e, para identificação diagnóstica de doenças no Brasil, tem-se 241 exames por imagem classificados como “disponíveis a população” (usuários do serviço público ou de saúde suplementar) (PARDINI, 2014). Cada exame pode ser aplicado em uma ou mais regiões do corpo humano, e para cada exame-região há diversos diagnósticos possíveis. Dado o grande número de doenças para classificação, criar sistemas especialistas para cada diagnóstico se apresenta como um grande desafio pra a comunidade de pesquisa em computação.

A Sociedade Brasileira de Computação (SBC) em seu II Seminário sobre Grandes Desafios da Computação no Brasil (2009), apresentou cinco grandes desafios da computação à serviço da sociedade, sendo dois deles relacionados com saúde, referenciados como S1 – Atividades de Medicina de Serviços de Saúde mediados por Computador; e S5 – Saúde e Software Onivalente, competência nacional de desenvolvimento de aplicações e serviços confiáveis na área de saúde. Neste, foi apresentado como um dos caminhos para atingir objetivos relevantes à sociedade, como saúde onivalente, “*Envolver a Indústria nas discussões, como forma de obter respaldo e validação*” (SBC, 2009)

Algoritmos para aprendizado supervisionado de classificação n-ária de imagens são amplamente utilizados nas pesquisas deste domínio, é comum a escolha de um tipo de exame de uma determinada região do corpo e uma patologia específica para o estudo da tarefa. Para cada patologia, segue-se um ciclo de trabalho que pode ser resumido nos seguintes passos: (i) aquisição de uma coleção de imagens com a patologia estudada; (ii) classificação adequada de imagens saudáveis; (iii) pré-processamento das imagens; (iv) subciclo de experimentação com treino, testes, verificação de resultados e ajustes de parâmetros do algoritmo e (v) geração de modelo de predição final baseado no resultado mais eficaz. Este ciclo pode ser percebido em trabalhos como Hsu, Chang & Lin (2003), Abramoff & Niemeijer (2008), Cunha e outros (2016) e Mendes (2017).

Ainda sobre de algoritmos de classificação, Piatetsky-Shapiro (1990) indica ser uma boa prática o uso de restritores (*constraints*), gerados a partir de conhecimento de domínio, para a tarefa de geração de conhecimento a partir de um

conjunto de dados (*dataset*), a fim de diminuir o espaço de busca. Os trabalhos encontrados neste domínio porém, comumente usam o conhecimento prévio como indicador de atributos e dimensões que devam ser adicionados ao *dataset*, aumentando o espaço de busca. Podemos observar um exemplo dessa prática no trabalho de Moskovitch & Shahar (2009). Dado que o espaço de busca possui relação direta com o tamanho do dataset necessário para extração de conhecimento e a acurácia obtida pelo sistema (em contagem de casos pré-classificados para aprendizado supervisionado) (WITTEN et al., 2016), tal prática aumenta a complexidade da tarefa e o trabalho computacional para sua realização.

Outra classe de algoritmos muito encontrada na literatura são os relacionados a tarefa de segmentação semântica de imagens, como apresentado em Bleau & Leon (2000), Jan (2006) e Félix (2007). Esses têm como objetivo separar as estruturas anatômicas apresentadas nas imagens de exames para facilitar a extração ou identificação de características na imagem que possam ser usadas para descrever tal estrutura (chamados de descritores), como por exemplo, dimensões, massa, densidade e coloração — ou apenas como mais uma etapa de pré-processamento para um algoritmo de classificação. Na Figura 3 é apresentado o exemplo de um exame de ressonância magnética 3D de crânio já com as estruturas anatômicas segmentadas

Algoritmos de visão computacional nos permitem, entre outras realizações, a extração de atributos do domínio importantes para apoio ao ato médico diagnóstico.

Na medicina diagnóstica existe amplo conhecimento formal relacionando evidências obtidas através de exames com enfermidades presente no paciente ou com alto grau de probabilidade de ocorrência em determinado período de tempo. São exemplos de tal conhecimento os trabalhos de Nover (2000), onde são listados e exemplificados evidências de patologias identificáveis através de exames de fundoscopias, Webb & Higgins (1999) que executa trabalho semelhante com imagens de tórax obtidas através de radiologia, tomografia computadorizada e ressonância magnética, identificando marcadores em pulmões e coração, e Wolff & Johnson (2009), que apresenta marcadores para imagens fotográficas dermatológicas.

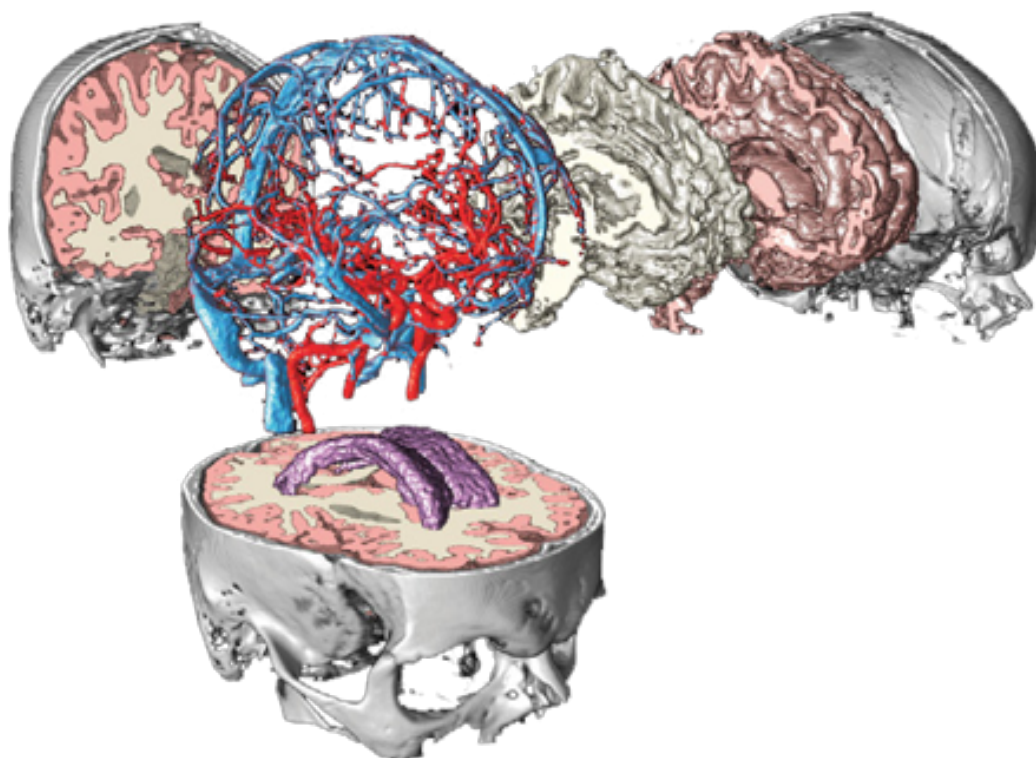


Figura 3 - Exemplo de segmentação de imagem 3D (RSNA, 2017)

Apesar destes fatos, nos trabalhos referenciados por essa dissertação não foram encontrados aqueles que se aproveitem diretamente deste conhecimento para a realização da tarefa de apoiar o ato médico diagnóstico, predizendo, classificando ou identificando características de interesse.

1.2

O Problema

Dado o grande número de diferentes exames diagnóstico por imagem, assim como o de doenças identificáveis por imagem (PARDINI, 2014), a criação de um grande conjunto de sistemas com alto grau de especialização em cada sistema pode ser uma abordagem custosa, de difícil utilização e pouco prática tanto para sua construção quanto para sua utilização. O desenvolvimento tradicional de classificadores através de modelos gerado com técnicas de aprendizagem de máquina necessita, para a sua construção, de uma base de dados com casos

previamente classificados. Em muitos casos a aquisição e configuração de tais bases demanda muito trabalho ou simplesmente não existem.

O problema em estudo é como facilitar o desenvolvimento de ferramentas para apoio ao diagnóstico médico de exames por imagem utilizando estratégias que:

1. Maximize sua utilidade para o domínio de medicina;
2. Facilite para desenvolvedores e engenheiros de software sua evolução e adaptação em diferentes subdomínios;
3. Facilite o desenvolvimento de sistemas que apoiem o diagnóstico de exames em *batch* (lote, batelada);
4. Aproveite diretamente o conhecimento existente no domínio da medicina diagnóstica, permitindo usá-los para gerar ou aprimorar sistemas computacionais classificadores de imagens.

1.3

Limitações nos Trabalhos Referenciados

Embora seja possível encontrar soluções na literatura que possibilitam o projeto ou implementação de sistemas classificadores de exames médicos por imagem (PIATETSKY-SHAPIRO, 1990; LIN; DEMNER-FUSHMAN, 2006; FÉLIX, 2007; MINA et al., 2007; ABRAMOFF et al., 2008; AYED et al., 2011; MACEDO et al., 2012; CUNHA; VENIERIS, 2015; CUNHA et al., 2016; TOENNIES; KÖNIG; GLOGER, 2016; GARDEZI; ELTOUKHY; FAYE, 2017; MENDES, 2017), nos trabalhos referenciados por essa dissertação, não foram encontrados aqueles que implementem algoritmos que se utilizem de conhecimento do domínio médico para inferir sobre as imagens sem a necessidade da etapa de treinamento para geração de modelo de classificação. Os trabalhos referenciados focam em doenças específicas, gerando soluções com alto nível de especialização, e não em abordagens que possibilitem classificar tais exames em escalares, possibilitando algum tipo de ordenação por unidades de medida definidas e interessantes para o contexto, como gravidade, urgência ou probabilidade de possuir de alguma patologia.

Outra limitação observada se associa ao alto grau de especialização destes sistemas. Tais implementações não possuem como objetivo a utilização de vários modelos para configuração dos classificadores, atendendo à diversos contextos em

um mesmo sistema. Os sistemas implementados observados são associados à um único diagnóstico.

1.4

Solução Proposta

Como solução ao problema identificado na seção 1.2 é proposta a construção de uma arquitetura de software e o desenvolvimento de um sistema no domínio de classificação de imagens de exames diagnósticos que facilite a evolução e configuração para as diversas áreas da medicina e tipos de exames por imagem, assim como a definição de uma abordagem que permita a generalização do problema de classificação de doenças.

A arquitetura proposta segue a *Entity-Component-System*. Este tipo de arquitetura visa maximizar a modularidade, modificabilidade, reutilização e usabilidade das interfaces do sistema (FISCHBACH; WIEBUSCH; LATOSCHIK, 2017). Explicada em detalhes na seção 2.5 deste documento, uma arquitetura deste tipo nos permite que, dado um ciclo processual base, se escolha dinamicamente quais métodos serão executados e quais atributos serão considerados durante o ciclo. Esta alternativa se mostrou interessante para a solução, os motivos serão esclarecidos a posteriori.

1.4.1

Abordagem e Estratégia de Aprendizado de Máquina

Ainda com o objetivo de reduzir os desafios apresentados na seção 1.2, este trabalho indica a utilização das diversas técnicas de aprendizado de máquina (ML) para alcançar os objetivos desejados e não somente os classificadores n-ários. Esta abordagem sugerida será explicada a seguir.

A estratégia comumente utilizada nos sistemas de ML (como os mencionados na seção 1 deste documento) é a de, através dos algoritmos de ML, apresentar ao sistema um conjunto de imagens com características de uma determinada doença (Figura 4-A), imagens saudáveis (Figura 4-B) e imagens com características de outras doenças (Figura 4-C) devidamente rotuladas, com a finalidade de se obter um modelo descritor das características da doença pesquisada. Com o modelo

obtido nessa fase de aprendizado é possível fazer a classificação de uma imagem nova.

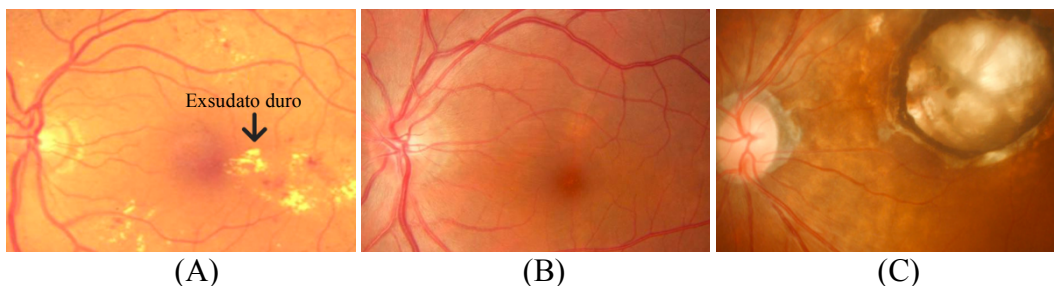


Figura 4 - Exemplos de imagens de fundo de olho com exsudatos duros (A) saudáveis (B) e (C) com outra doença não classificada

A classificação descreve se a imagem possui uma característica pesquisada, apresentando também a acurácia de tal sistema.

Este trabalho propõe desenvolver um processo onde seja possível, além de executar algoritmos da abordagem tradicional de ML, adicionar ao fluxo de execução do processamento de exames algoritmos que executem a tarefa de reconhecimento de anomalias de qualquer natureza presentes em uma imagem de exame, classificando-a com um escalar comparável a um valor padrão de imagem saudável. Essa abordagem se mostra bastante interessante pelo fato de permitir a utilização deste trabalho como base para a criação de sistemas automatizados de apoio à priorização de atendimento de pacientes, ampliando o conceito de apoio ao diagnóstico, comumente associado ao significado restrito da palavra (laudar ou identificar uma doença ou exame), para o conceito amplo e original (descobrir, sinalar, apontar) (AURÉLIO, 2010).

1.5

Coleta de Informações para Estabelecimento da Base de Pesquisa

Na pesquisa preliminar realizada para este trabalho, cinco tópicos foram selecionados para estudo e estabelecimento de uma base de conhecimento primária, são eles:

1. Exemplos de algoritmos de classificação e segmentação de imagens, assim como exemplos de desenvolvimentos de sistemas que utilizem

tais algoritmos. Foram considerados os exemplos contidos em 27 trabalhos selecionados pelo autor por sua relevância técnica ou por tema associado a subdomínios variados do campo da medicina diagnóstica. São eles: (HOFFMAN; RICHARDS, 1984; RUCK et al., 1990; OTHMAN; DORIZZI; GARCIA-SALICETTI, 2016; FILHO; NETO, 1999; BLEAU; LEON, 2000; HSU; CHANG; LIN, 2003; JAN, 2006; RASMUSSEN; WILLIAMS, 2006; FÉLIX, 2007; ABRAMOFF et al., 2008; AYED et al., 2011; BOTTOU, 2010; LECUN; KAVUKCUOGLU; FARABET, 2010; MACEDO et al., 2012; DIAS et al., 2012; ZHOU et al., 2012; OREIFEJ; LIU, 2013; CUNHA; VENIERIS, 2015; CUNHA et al., 2016; EL-BAZ; JIANG; SURI, 2016; TOENNIES; KÖNIG; GLOGER, 2016; WITTEN et al., 2016; GREENSPAN; VAN GINNEKEN; SUMMERS, 2016; MIRZAEI; ADELI, A.; ADELI, H., 2016; WANG et al., 2016; GARDEZI; ELTOUKHY; FAYE, 2017; MENDES, 2017).

2. Desenvolvimento de um sistema classificador exemplo, utilizado como primeiro experimento com classificadores de imagens no domínio da medicina diagnóstica. Tal trabalho gerou quatro artefatos resultantes. São eles:
 - a. Uma monografia escrita para o curso de Aprendizagem de máquina na PUC Rio de título “Aprendizagem de máquina na tarefa de classificação de doenças a partir de imagens de fundo de olho” (CUNHA; VENIERIS, 2015), que utilizou algoritmo classificador de imagem baseado em Support Vector Machines (SVM) para identificar imagens de exames de fundoscopia que possuam a presença exsudatos duros, marca identificadora de retinopatia diabética.
 - b. Um artigo publicado no XV Congresso Brasileiro de Informática em Saúde (CBIS 2016) de título “Uso de aprendizagem de máquina para classificação de imagens de fundo de olho” (CUNHA; VENIERIS, 2015), utilizando a mesma abordagem utilizada na monografia de Cunha & Venieris (2015).

- c. Um conjunto de dados (*dataset*) com imagens de fundoscopia devidamente rotuladas nas seguintes 3 classes: saudáveis, com retinopatia diabética e com outras doenças. Este *dataset* é artefato vital para o algoritmo de aprendizagem de máquina, que o utiliza para executar a tarefa de aprendizado supervisionado, gerando um modelo a ser utilizado para classificação de novas imagens. Tal *dataset* foi criado a partir de uma coleção de 32.000 exames contendo 140.000 imagens laudadas, gentilmente cedidas pelo Instituto de Olhos de Duque de Caxias (INODUC).
 - d. Um sistema de classificação de imagens que se utilize do modelo gerado nos trabalhos de Cunha & Venieris (2015) e Cunha e outros (2016) para testes e experimentações.
3. Execução de teste prático com uma médica oftalmologista e o sistema classificador para apoio ao ato diagnóstico, desenvolvido nos trabalhos de Cunha & Venieris (2015) e Cunha e outros (2016), a fim de observar sua interação com tal sistema.
 4. Limitações legais para o uso prático de sistemas classificadores e considerações relevantes para o desenvolvimento de sistemas dessa natureza, considerando tais limitações. Esta parte do estudo está descrita em detalhes na seção 7.2 deste documento.
 5. Busca por sistemas que sejam comercialmente utilizados e utilizem técnicas de classificação ou segmentação automatizada de imagens. Neste foram considerados os sistemas desenvolvidos pelas empresas que possuem ou possuíram certificação para Sistemas Eletrônicos em Saúde emitidos pela Sociedade Brasileira de Informática em Saúde em conjunto com o Conselho Federal de Medicina (SBIS-CFM), no ano de 2016, filtrando sistemas de apoio ao ato médico que possuíssem funções de classificação e/ou segmentação de imagens de exames (SBIS, 2016).

As conclusões do autor, resultantes destas pesquisas foram as seguintes:

- Sistemas de apoio à segmentação semântica e quantificadores de atributos de imagens são utilizados em medicina diagnóstica,

normalmente fornecidos em conjunto ou embarcados no hardware capaz de realizar o exame. São exemplos destes sistemas Osiris 4.1 (OTHMAN; DORIZZI; GARCIA-SALICETTI, 2016) e SDEP (FÉLIX, 2007)

- A utilização de classificadores em sistemas de exame diagnóstico por imagem é incomum, não foi encontrado nos sistemas listados como ora certificados pela (SBIS-CFM) um sistema aplicado em centros de diagnóstico clínico no Brasil com esta finalidade.
- A experiência de uso de sistema de classificação com o médico oftalmologista apresentou um ganho bastante baixo ao trabalho de diagnóstico, segundo opinião da Dra. Monica Almeida Venieris (CRM: 520531-RJ), “*não há motivos para se ter um sistema que classifique uma imagem no mesmo tempo e momento que o médico, não há ganhos nisso*”.
- Sistemas de apoio diagnóstico aplicados devem trabalhar fora da área do que é considerado ato médico. As fronteiras e especificações do que é ato médico e suas implicações são abordadas na seção 7.2 deste documento.

Após estas conclusões, foi pesquisado que tipos de projetos médicos poderiam se beneficiar de sistemas automatizados de diagnóstico. Foram encontrados diversos projetos baseados em campanhas de exames em larga escala, por destaque cita-se o programa *De olho na Visão* do governo do estado de Goiás (GEGO, 2016), que entre 2014 e 2016 realizou 2 milhões de fundoscopias. Outras campanhas podem ser citadas, como a *Campanha de Prevenção ao Glaucoma* (PROGRESSO, 2015), de periodicidade anual, ministrada pela Associação Mato-Grossense de Oftalmologia (AMO), que apenas na edição de maio de 2015 realizou 3.000 fundoscopias em um único dia, e o *Mulheres de Peito* de São Paulo, que com as *Carretas da Mamografia* (GESP, 2016) atende mulheres a partir de 50 anos sem a necessidade do pedido médico. Este último realizou no período de 1 ano 103.400 exames, sendo 98.644 mamografias, 4.266 ultrassonografias e 530 biópsias, direcionando 1.343 mulheres aos centros de referência de combate ao câncer.

Tais campanhas operam em duas etapas, na primeira há a coleta das imagens e/ou material, por médicos, estagiários, enfermeiros ou leigos orientados no

processo do exame e na segunda um grupo de médicos especialistas é contratado para laudar cada um dos exames. Após análise das informações coletadas o enfoque do trabalho foi definido, assim como a estratégia apresentada na seção 1.4.1.

1.6

Contribuições Esperadas

O principal componente para solução do problema é o desenvolvimento de um sistema com um núcleo base que contém os seguintes itens:

- A definição do ciclo de execução do processo proposto;
- Um framework para definição do padrão de desenvolvimento das entidades e componentes (ECS), assim como protocolos de comunicação com as demais partes do sistema;
- Entidades e componentes que executem o trabalho para a prova de conceito, com ênfase em componentes que se utilizem de conhecimento do domínio médico-diagnóstico para a classificação de imagens;
- A definição de uma estratégia de geração de dados para avaliação da qualidade e relevância do apoio diagnóstico oferecido considerando as motivações explicitadas neste documento.

1.7

Organização da Dissertação

Esta dissertação é estruturada como segue:

- No Capítulo 2 é apresentada a fundamentação teórica necessária para a execução e entendimento do trabalho;
- No Capítulo 3 são descritos alguns dos trabalhos relacionados;
- No Capítulo 4 é apresentada a solução proposta para o problema apontado pelo trabalho;
- No Capítulo 5 há a Conclusão do trabalho, contribuições e trabalhos futuros.

2 Fundamentação Teórica

Esta seção descreve as principais características dos conceitos necessários para o entendimento do trabalho proposto. Em primeiro lugar será abordado o processo **KDD** (*Knowledge Discovery in Databases*) apresentando o conceito e as etapas estabelecidas para o processo. Em seguida serão apresentados os conceitos de **Aprendizagem de Máquina**, com uma breve definição formal de o que é aprender e a exposição de um trabalho que se utiliza de tal técnica. Em terceiro lugar serão apresentados os principais conceitos de **Visão Computacional** assim como o fluxo de trabalho necessário para a realização do processamento de imagens utilizando esta técnica, em seguida será apresentado uma breve definição de **Deteção de Anomalias**, descrevendo do que se trata e como tal técnica pode contribuir para identificação imagens de exames médicos que possuam características anômalas. Será apresentado também o padrão de arquitetura **Entity-component-system**, em sua forma e pontos de interesse com exemplos para concretização de seus conceitos. O sexto conceito abordado será o de **Frameworks**, importante tema de engenharia de software, apresentando suas definições. Por último o conceito e definições de **Arquitetura de Software** e como esse conceito será utilizado no trabalho.

2.1 KDD

O processo KDD (*Knowledge Discovery in Databases*) define seis etapas para obtenção de massas de dados e tratamento dos mesmos para extração de informação e obtenção de conhecimento (ALMEIDA, 2004). A Figura 5 mostra as fases do processo KDD apresentadas por Fayyad e outros (1996). A seguir será descrito cada etapa, seu objetivo e importância.

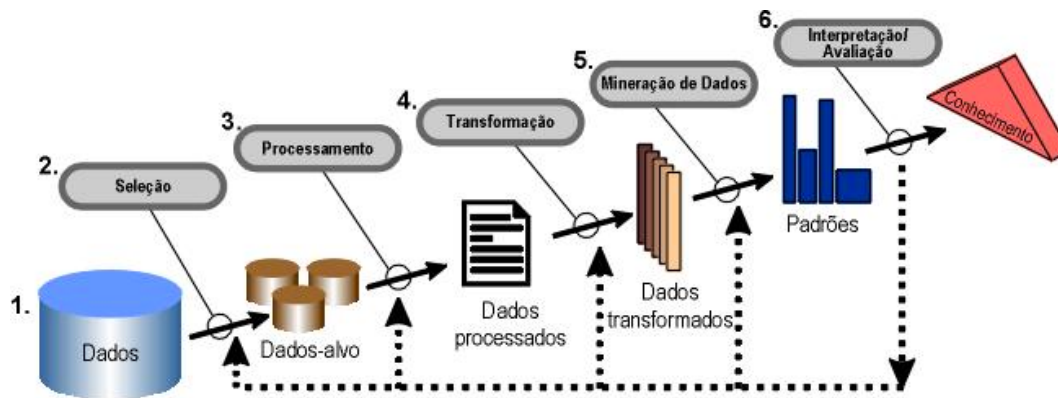


Figura 5 - Fases do processo KDD (FAYYAD et al., 1996)

A primeira etapa é a obtenção dos **dados**, adequando-o para que seu acesso seja facilitado para o processo seguinte, nesta etapa os dados podem tanto ser formatados (como por exemplo tabelas em bases de dados) como não formatados (como arquivos de texto em qualquer formato, arquivos imagens ou de áudio).

A segunda etapa é a de **seleção**, onde são definidos os dados realmente importantes para a extração de informações e posteriormente do conhecimento, descartando os dados não necessários para a obtenção do conhecimento desejado.

A terceira etapa é a de **processamento**, o objetivo desta etapa é reduzir a complexidade do problema. Nesta etapa é efetuada a limpeza dos dados, identificação de inconsistências, caso desejável, a eliminação de amostras inconsistentes ou aberrantes (*outliers*) e os atributos mais significativos são selecionados, reduzindo a dimensionalidade do problema e evitando que variáveis correlatas sejam utilizadas em conjunto na análise.

A quarta etapa é a de **transformação**, onde se busca codificar e transformar os dados para facilitar o processo de mineração. Seguem alguns exemplos de transformação:

- Transformação de variáveis contínuas em categóricas, como por exemplo faixas de valores dentro e fora de uma classificação. Na Figura 6 é apresentado um exemplo de transformação de valores contínuos em categóricos. Nesse exemplo os valores contidos na faixa de 0,00 à 5,90 recebe um rótulo 'A', de 6,00 à 9,00 um rótulo 'B' e de 10,00 à 99,9 um rótulo 'C'.

Classe	Min	Máx
A	0,00	5,90
B	6,00	9,00
C	10,0	99,9

Figura 6 - Exemplo de transformação de valores contínuos para categóricos

- Normalização de valores variáveis, onde por exemplo, aplica-se um cálculo de transformação de valores para que estes se mantenham dentro do intervalo [0,1].
- Reflexão de dados simétricos, onde quando há a necessidade de geração de mais dados para os processos seguintes e existe algum conceito de simetria, os dados podem ser duplicados com a aplicação do cálculo de simetria para a sequência duplicada. Como exemplo podemos considerar o trabalho de Da Silva Moreira e outros (2017), onde foram usados dados de captura de movimento dos braços em um algoritmo de aprendizagem de máquina para compreensão de movimentos. Os dados de movimento do braço esquerdo podem ser refletidos para que se produzam novos dados representando movimentos do braço direito, duplicando assim a base de dados para aprendizagem.

O artefato resultante dessa fase é um *Data Warehouse* (DW). DWs são bancos de dados voltados para consulta, geralmente possuindo dados consolidados, de caráter histórico, não operacionais e provenientes de diversas fontes. Com um DW montado, tem-se um repositório central com todos os dados necessários para a exploração, já selecionados, limpos e transformados.

A quinta etapa é a **Mineração de Dados** (*Data Mining*), responsável por extrair informações implícitas e potencialmente úteis a partir dos dados. Essas informações serão utilizadas pelos usuários ou por outros sistemas para a geração de conhecimento e, principalmente, para tomada de decisões estratégicas. A mineração de dados é proveniente da técnica de aprendizagem de máquina

(*machine learning*) e atualmente diversas técnicas são usadas para realizar essa tarefa, não sendo elas objeto de estudo deste documento.

A sexta e última etapa é a **interpretação** ou **avaliação**. É a fase onde o conhecimento é gerado, resultando em artefatos como modelos de classificação de padrões ou preditor de séries temporais. Com esses modelos pode-se inferir informações sobre novos dados, como por exemplo, a que classe ou tipo pertence.

2.2

Aprendizagem de Máquina

A área de aprendizagem de máquina (ML) é um subdomínio inteligência artificial (*Artificial Intelligence* - AI) e é o campo de estudo que proporciona aos computadores a habilidade de aprender a resolver um determinado problema sem ser explicitamente programado para tal tarefa (SIMON, 2013).

Para que a definição anterior faça sentido é necessário também definir o significado de aprender. Um programa de computador aprende através de uma experiência (E), a respeito de um conjunto de tarefas (T), mensuráveis por uma métrica (P) se o seu desempenho na tarefa (T), medida pela métrica (P) melhora com a experiência (E) (MITCHELL, 1998a). Algoritmos de aprendizado de máquina são utilizados amplamente, com destaque nas áreas de processamento natural de linguagem, visão computacional e mineração de dados.

O conceito acima é demonstrado em no trabalho de Cunha e outros (2016). Para a tarefa de se criar um classificador de imagens de fundoscopias que possuem exsudatos duros (marca patogênica identificável no exame de fundoscopia) foram executadas as 3 etapas do processo de desenvolvimento do sistema de ML. Na primeira etapa, a fase de treino, um conjunto de imagens devidamente rotuladas são submetidas ao algoritmo de ML para geração de um modelo da classe em estudo. Na segunda etapa, a fase de testes, um outro conjunto de imagens devidamente rotuladas são comparadas com o modelo e a classificação do sistema comparada com o rótulo, obtendo-se com este processo valores como acurácia, sensibilidade e especificidade. Na terceira etapa uma imagem sem rótulo é submetida ao sistema, que a classifica.

Os algoritmos de aprendizagem o fazem seguindo um paradigma de aprendizado, estes podem ser divididos em dois grandes grupos, os de aprendizado supervisionado e os de aprendizado não supervisionado.

Os algoritmos de aprendizado não supervisionado operam analisando exemplos fornecidos e tentam determinar se alguns deles podem ser de alguma forma agrupados. Após a definição dos agrupamentos é necessária uma intervenção humana para determinar o que cada agrupamento significa no contexto do problema que está sendo analisado (CHEESEMAN et al., 1990). Esta abordagem não se mostrou promissora para a resolução do problema abordado por esta dissertação pois não conforma com a abordagem proposta na seção 1.4.1 deste documento, é comumente usado para agrupar exemplos em classes quando tais classes são desconhecidas ou o agrupamento é mais importante do que a rotulação (CHEESEMAN et al., 1990), o que não é verdade no problema em questão.

Já os paradigmas de aprendizado supervisionado buscam rotular exemplos em uma classe à partir de um conjunto de classes conhecidas (MONARD et al., 2003), objetivo que conforma com a abordagem proposta na seção 1.4.1 deste documento, concluindo-se que a arquitetura desenvolvida deve suportar a utilização de algoritmos baseados em paradigmas de aprendizagem supervisionada.

Alguns exemplos de paradigma de aprendizado supervisionado são: Simbólico, Estatístico, Baseado em Exemplos, Conexionalista e Genético. A seguir uma breve descrição destes.

- **Simbólico:** Os sistemas de aprendizado simbólico buscam aprender construindo representações de um conceito através de exemplos e contraexemplos deste conceito, tipicamente representados em forma de alguma expressão lógica, árvore de decisão, regras ou rede semântica.
- **Estatístico:** Os sistemas de aprendizado estatístico utilizam modelos estatísticos para encontrar uma boa aproximação do conceito induzido, assumindo alguma forma de modelo paramétrico, e então encontrando valores apropriados para os parâmetros do modelo a partir dos dados.

Pode-se tomar como exemplo um classificador que assume que as classes podem ser expressas como combinação linear dos valores dos atributos, e então procura uma combinação linear

particular que fornece a melhor aproximação sobre o conjunto de dados.

Dentre os métodos estatísticos, destacam-se os de aprendizado Bayesiano, que utilizam um modelo probabilístico baseado no conhecimento prévio do problema, o qual é combinado com os exemplos de treinamento para determinar a probabilidade final de uma hipótese (MITCHELL, 1998a). Alguns autores tem considerado Redes Neurais como métodos estatísticos paramétricos, uma vez que treinar uma Rede Neural geralmente significa encontrar valores apropriados para pesos e bias pré-determinados.

- **Baseado em Exemplos:** A ideia deste paradigma é classificar um exemplo buscando em uma base de conhecimento outro exemplo similar e assumir que o novo exemplo será de mesma classe. Esse tipo de sistema de aprendizado é também chamado de *lazy* (preguiçoso). Sistemas *lazy* necessitam manter os exemplos na memória para classificar novos exemplos, em oposição aos sistemas *eager* (ansioso), que utilizam os exemplos para induzir um modelo, descartando-os após a utilização (AHA, 1997). Saber quais exemplos devem ser memorizados por um indutor *lazy* é a chave para o sucesso na geração deste classificadores que utilizam este paradigma, deve-se buscar reter apenas aqueles mais representativos do problema. Aha Kibler & Albert (1991) descrevem algumas estratégias de decisão para a retenção em memória de um exemplo. Nearest Neighbours (RUIZ, 1986) e Case-Based Reasoning (CBR) (WATSON; MARIR, 1994) são alguns exemplos de algoritmos que utilizam este paradigma.
- **Conexionista:** Um exemplo claro deste paradigma são os algoritmos de Redes Neurais (construções matemáticas simplificadas inspiradas no modelo biológico do sistema nervoso) esta envolve unidades altamente interconectadas, por esse motivo o nome conexionismo é utilizado para descrever esta área de estudo. Redes Neurais possuem grande potencial na resolução de problemas que requerem intenso processamento sensorial, tal como visão e reconhecimento de voz (BRAGA; CARVALHO; LUDERMIR, 2000).

- **Genético:** Um classificador genético consiste de uma população de elementos de classificação que competem para fazer a predição. Elementos que possuem uma performance fraca são descartados, enquanto os elementos mais fortes proliferam, produzindo variações de si mesmos. Este paradigma possui uma analogia direta com a teoria de Darwin, na qual sobrevivem os mais bem adaptados ao ambiente (GOLDBERG; HOLLAND, 1988).

Como dito anteriormente, a partir dos paradigmas de aprendizado foram desenvolvidos os algoritmos de aprendizagem de máquina. Alguns destes são brevemente descritos a seguir:

- *Support Vector Machines* (SVM). Uma máquina de vetor de suporte (SVM em livre tradução) funciona mapeando dados para um espaço de variável altamente dimensional, para que os pontos de dados possam ser categorizados mesmo quando não forem linearmente separáveis de outra forma. Após um separador entre as categorias ser localizado, os dados são transformados de modo que o separador possa ser desenhado como um hiperplano. Com um modelo de dados possuidor deste separador desenhado as características de novos dados de entrada podem ser utilizadas para prever o grupo ao qual um novo registro deve pertencer. A função matemática usada para a transformação é conhecida como a função *kernel*. Existem alguns tipos de função *kernel*, como linear, polinomial, RBF (função de base radial) e curva sigmoide (PLATT, 1998; KEERTHI et al., 2001).
- *Pairwise*. Os algoritmos SVM comumente são binários. Classificadores *Pairwise* (também conhecido como Round Robin) foram apresentados por Hastie e Tibshirani (1998) como uma estratégia para combinar classificadores binários em soluções multiclases, onde para q classes são necessários: $q(q-1)/2$ classificadores (HASTIE; TIBSHIRANI, 1998).
- *Naïve Bayes*. Baseado no teorema de Bayes, a técnica de classificação supõe independência entre os preditores, isto é, assume que a presença de uma característica particular em uma classe não está relacionada

com a presença de qualquer outro recurso. Por exemplo, uma amostra pode ser considerada uma planta se for verde e possuir vários ramos. Mesmo que esses recursos dependam uns dos outros ou da existência de outras características, todas estas propriedades contribuem de forma independente para a probabilidade de que a amostra é uma planta, este é o motivo do termo ‘*Naive*’ (ingênuo). O modelo Naive Bayes é fácil de construir e particularmente útil para grandes conjuntos de dados e classificações multi-classe (JOHN; LANGLEY, 1995).

- MLP (*Multilayer Perceptron*). As MLP (Redes Neurais Perceptron de Multi-camada) são baseadas em redes neurais artificiais. As RNA (ou ANN em inglês) são modelos matemáticos que se assemelham às estruturas neurais biológicas e que têm capacidade computacional adquirida por meio de aprendizagem e generalização (BRAGA; CARVALHO; LUDERMIR, 2000). Esses modelos procuram se assemelhar ao sistema nervoso dos seres vivos e sua capacidade de processar informações. É definida como uma máquina adaptativa contendo um processador dotado de paralelismo distribuído, constituído de unidades de processamento simples com a propensão natural para armazenar conhecimento experimental e torná-lo disponível para uso. Se assemelha ao cérebro, onde o conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de aprendizagem e forças de conexão entre neurônios, conhecidas como pesos sinápticos, são utilizadas para armazenar o conhecimento adquirido (HAYKIN et al., 2001). Perceptron é exemplo de um RNA de camada simples, isto é, compostas por neurônios computacionais, com uma camada de entrada e uma camada de saída. A sua implementação base consiste de um único neurônio com pesos sinápticos e viés (bias) ajustáveis. As redes de múltiplas camadas (MLP) distinguem-se das redes de camada simples pelo número de camadas intermediárias, aquelas entre a camada de entrada e a de saída. Essa arquitetura possui uma ou mais camadas ocultas. Segundo HAYKIN E OUTROS (2001), a função dos neurônios ocultos é intervir entre a camada de entrada externa e a saída da rede de maneira

útil. Adicionando-se uma ou mais camadas ocultas, tornamos a rede capaz de extrair estatísticas de ordem elevada. Em sua implementação base os 3 tipos de camadas da MLP e suas funções são:

- Camada de entrada, não-computacional, nela não há processamento, responsável pela recepção e propagação das informações de entrada para camada seguinte;
- Camadas intermediárias, uma ou mais camadas ocultas, compostas por nós. São camadas computacionais (que efetuam processamento), nelas são transmitidas as informações por meio das conexões entre as unidades de entrada e saída. Essas conexões guardam os pesos que serão multiplicados pelas entradas, garantindo o conhecimento da rede.
- Camada de saída: Camada composta por neurônios computacionais que recebem as informações das camadas ocultas fornecendo a resposta.

O algoritmo de treinamento mais utilizado em modelos MLP é o Backpropagation, que se baseia na aprendizagem por correção de erros. O algoritmo de Backpropagation é um tipo de aprendizado supervisionado, quando o valor de saída é gerado o erro é calculado e seus valores são retro propagados para entrada, os pesos são ajustados e os valores são novamente calculados. A habilidade dos neurônios ocultos extraírem estatísticas de ordem elevada é particularmente valiosa quando o tamanho da camada de entrada é grande. (RUCK et al., 1990).

Os componentes de classificação desenvolvidos em conformidade com a arquitetura proposta devem possuir uma função f , que receba um modelo M , e um exemplo E , retornando um rótulo (*label*) L , com a assinatura $f(M, E) = L$.

2.3

Sistemas de Visão Computacional

Visão computacional é a construção de descrições explícitas e claras dos objetos em uma imagem (BALLARD; BROWN, 1982). Trata explicitamente da

obtenção e manipulação dos dados de uma imagem e do uso deles para diferentes propósitos.

Um sistema de Visão Computacional é constituído por processos capazes de adquirir, processar e interpretar imagens correspondentes de cenas obtidas em ambientes reais e é constituído por processos como aquisição de imagens, pré-processamento, segmentação, extração de atributos, reconhecimento e interpretação (FILHO; NETO, 1999). Um exemplo é apresentado na Figura 7, e sua descrição a seguir.

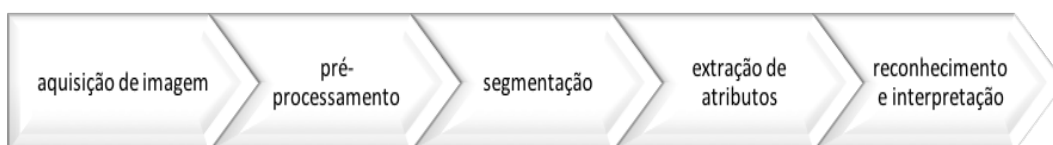


Figura 7 - Sistema típico de visão computacional (FILHO; NETO, 1999)

2.3.1

Aquisição de Imagens

A etapa de aquisição tem como função converter uma imagem em uma representação numérica adequada para o processamento digital subsequente. Este bloco compreende dois elementos principais. O primeiro é um dispositivo físico sensível à uma faixa de energia no espectro eletromagnético (como raio X, ultravioleta, espectro visível ou raios infravermelhos), que produz na saída um sinal elétrico proporcional ao nível de energia detectado. O segundo, o digitalizador propriamente dito, converte o sinal elétrico analógico em informação digital, isto é, que pode ser representada através de bits 0s e 1s.

Esta etapa é finalizada com o armazenamento das imagens em meio adequado para o processamento.

2.3.2

Pré-processamento

Melhoramento das características relevantes da imagem como normalização de histograma, minimização de ruídos, recorte de partes não relevantes e alteração de resolução e cores com o objetivo de facilitar as etapas seguintes.

2.3.3

Segmentação

A segmentação de imagens é o processo de dividir semanticamente uma imagem em múltiplas regiões. Utilizando complexas técnicas de inteligência artificial é efetuado um recorte para cada elemento da imagem, ferramenta de fundamental importância em qualquer sistema de Visão Computacional, o desempenho do sistema depende essencialmente do desempenho do processo de segmentação.

As técnicas usadas no processo de segmentação variam de acordo com as características e objetivos das imagens e do sistema. Algumas destas técnicas são: limiarização, segmentações baseadas em regiões, segmentação através de morfologia matemática, watershed, wavelet, contornos ativos e outras. A discussão de tais técnicas será alvo de trabalhos futuros.

2.3.4

Extração de Atributos

Esta é a etapa em que as características de interesse (atributos ou feições) dos objetos segmentados são analisadas e valoradas para, desta forma, facilitar a execução do processo de reconhecimento e interpretação. Na extração dos atributos podem ser utilizadas técnicas como: código de cadeia, aproximação poligonal, descritores de Fourier, descritores regionais, medidas de texturas, dentre outros. A discussão de tais técnicas será alvo de trabalhos futuros.

2.3.5

Reconhecimento e Interpretação

Esta é a etapa onde os objetos interpretados são rotulados de acordo com atributo ou feições de interesse identificados no processo anterior e agrupados em uma classe à qual aquele rótulo pertence definidos a priori. Neste processo podem ser utilizadas técnicas de reconhecimento de padrões, estatísticas, inteligência artificial, dentre outras.

Os componentes de segmentação desenvolvidos em conformidade com a arquitetura proposta devem possuir uma função f , que receba uma instância de uma classe I , representando um exame de imagem, e retorne uma coleção de instâncias de classes $[I']$ contendo em cada elemento uma subestrutura de I , com a assinatura $f(I) = [I']$.

2.4

Detecção de Anomalia

Algoritmos de detecção de anomalia assumem que, comparadas com instâncias normais, ocorrências anômalas são raras, e quando estas ocorrem, desviam significativamente do restante da amostra (HODGE; AUSTIN, 2004). No entanto, “significativamente” constitui um julgamento subjetivo que pode ou não classificar uma instância como anômala. Para a construção de um sistema de detecção de anomalias, inicialmente há de se definir o que é normal e seus limites. A anomalia irá se encontrar fora dos limites da normalidade, em seguida deve-se analisar das instâncias anômalas o que é ruído (exemplos imperfeitos causados por falha no processo de aquisição, transformação ou rotulação das classes) de o que é anomalia de fato. Dependendo do conjunto de amostras obtidos a complexidade da tarefa de classificação de anomalias se altera. Se há a garantia que não há ruídos nas amostras qualquer instância fora dos limites da normalidade podem ser classificadas como anomalias, caso contrário será necessária a aplicação de uma ou mais técnicas de redução e/ou identificação de ruídos para que a anomalia não se confunda com estes. Neste caso a crença (nível de certeza) da correta classificação é reduzida proporcionalmente à acurácia da operação de identificação de ruídos, podendo-se inclusive não ser possível a distinção entre ruído e anomalia. Na

Figura 8 há um exemplo de distribuição de dados, distinguindo dados normais, anomalias e ruídos.



Figura 8 - Exemplo de classificação ideal de instâncias normais, ruídos e anomalias.

A maioria dos algoritmos de detecção de anomalias usam alguma métrica para pontuar o nível da anomalia. Algumas destas técnicas são vizinho mais próximo, *clustering* ou inferência estatística (GUPTA et al., 2014; AGGARWAL, 2013). E nesses algoritmos, anomalias normalmente recebem uma pontuação muito maior do que os ruídos.

Não é objetivo deste trabalho o aprofundamento destas técnicas, porém a detecção e pontuação de amostras anômalas pode ser um importante aliado à geração de valores escalares para a ordenação de exames por métricas definidas por contexto, conforme explicado na seção 1.4.1 deste documento.

Os componentes de detecção de anomalias desenvolvidos em conformidade com a arquitetura proposta devem possuir uma função f , que receba um modelo (ou matriz de covariância) M e um exemplo de um exame E , retornando um escalar S que represente uma pontuação em relação à valores normais conhecidos e representados em M , com a assinatura $f(M, E) = S$.

2.5

Entity-Component-System

Entity-Component-System (ECS) é uma arquitetura de software comumente usada no desenvolvimento de jogos, sistemas de simulação e sistemas gerais que

requisitem alta modularidade, modificabilidade e reutilização (FISCHBACH; WIEBUSCH; LATOSCHIK, 2017). Um ECS segue o princípio da “composição ao invés de herança”, onde um sistema é composto por um conjunto de entidades coordenadas pelo objeto-sistema (o objeto que possui a responsabilidade de coordenar o sistema), onde cada entidade é composta por componentes, que adicionam atributos, comportamentos ou funcionalidades. Portanto, o comportamento de uma entidade pode ser alterado durante o tempo de execução simplesmente adicionando ou removendo componentes, adicionando maior flexibilidade na definição de entidades. Essa abordagem também elimina problemas de ambiguidade com que sofrem as hierarquias de herança profunda e vasta, que são difíceis de entender, manter e estender. Abordagens comuns de ECS são altamente compatíveis e, muitas vezes, combinadas com técnicas de design orientado por dados (FISCHBACH; WIEBUSCH; LATOSCHIK, 2017).

Zaks (2017) defende que ECS é um paradigma de programação, uma maneira de pensar. Em termos computacionais não provê nem mais nem menos poder que a orientação a objetos hierárquica. *“Não pense desta forma, quando esta forma de pensar não lhe proporcionar mais benefícios”* (apud NADEZHGIN, 2017) (livre tradução do autor). Para decidir quando usar ou não ECS, Zaks divide os sistemas computacionais em dois tipos:

- Computação pura, quando os sistemas são destinados à receber entradas e processar saídas. Tem um propósito único e seu ciclo de vida é limitado à própria computação. Exemplos deste tipo de sistema são compiladores, scripts, sistemas de funções gerais de entrada e saída, como cadastramento e organização de dados.
- Computação interativa, onde o sistema roda ao longo do tempo, além de receber entradas. Exemplos deste tipo de sistema são jogos e simuladores.

“Quando você precisa escrever um software de computação pura, a ECS não ajuda muito. O mantra principal da ECS é a separação de estado e comportamento. Quando temos computação pura, devemos passar o estado ao longo de funções concatenadas. Então, os paradigmas da programação funcional são a forma mais sensata de pensar aqui.

Quando fazemos computação interativa, temos estado acumulado ao longo do tempo e dependemos da entrada externa. Nessa situação, a ECS é útil. Pensando em dados primeiro, ajuda a construir uma lógica escalável, simples e desacoplada.” (livre tradução do autor)

Fischbach, Wiebusch & Latoschik (2017) ressaltam que ECS é uma reorganização ou extensão da programação orientada a objetos hierárquica padrão (OO). É possível à qualquer linguagem OO implementar um sistema ECS, e o ECS não inclui *overhead* (esforço operacional extra) significativo ao sistema.

No exemplo da Figura 9 é apresentada uma tentativa de se modelar o reino animal através de hierarquia. Para essa tarefa foram definidos níveis excludentes de características, como por exemplo vivíparos e ovíparos. Repare que para qualquer arranjo de características dentro de uma hierarquia, para resolução deste problema o resultado terá duplicação de classes, heranças que não fazem sentido ou classes específicas que replicam características parciais de outras classes. Em muitos sistemas com essas características observamos classes herdeiras diretas de classes de alto nível sem sucessores e com características replicadas em outras classes. Ainda no exemplo da Figura 9 é representado o ornitorrinco, que é um mamífero ovíparo com diversas características de aves aquáticas como membranas interdigitais e corpo hidrodinâmico, um exemplo claro das limitações da modelagem com herança. No exemplo do ornitorrinco qualquer decisão de hierarquia a partir do modelo proposto resultará em replicação de características. Como este é um mamífero ovíparo, esta característica (entre outras de caso semelhante) só poderá ser representada com a replicação de características, ora excludentes.

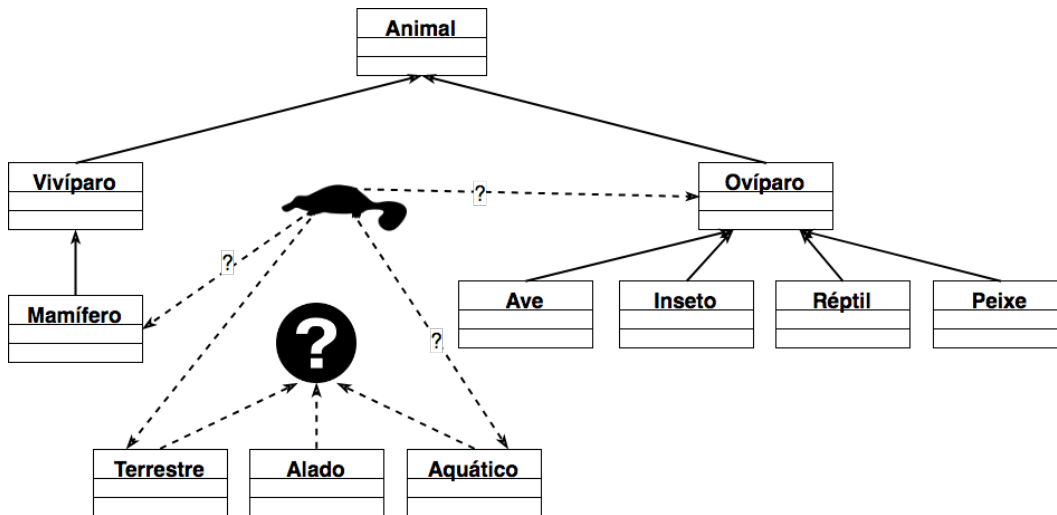


Figura 9 - Tentativa de modelar o reino animal utilizando hierarquia

Guedes (2008) resolve parcialmente este problema apresentando Herança Múltipla, onde uma subclasse pode herdar as características de duas superclasses. No exemplo do ornitorrinco apresentado acima, segundo Guedes, este poderia herdar as características dos animais vivíparos e ovíparos, porém esta solução resultaria em uma modelagem imprecisa e complicada, adicionando características indesejadas. No exemplo, um útero (presente no animais vivíparos), originalmente inexistente no ornitorrinco, é adicionado à este apenas para que se mantenha a modelagem com herança.

Resolvendo o mesmo problema com ECS podemos compor qualquer animal com suas características, sem replicações de códigos ou adição de características não utilizadas.

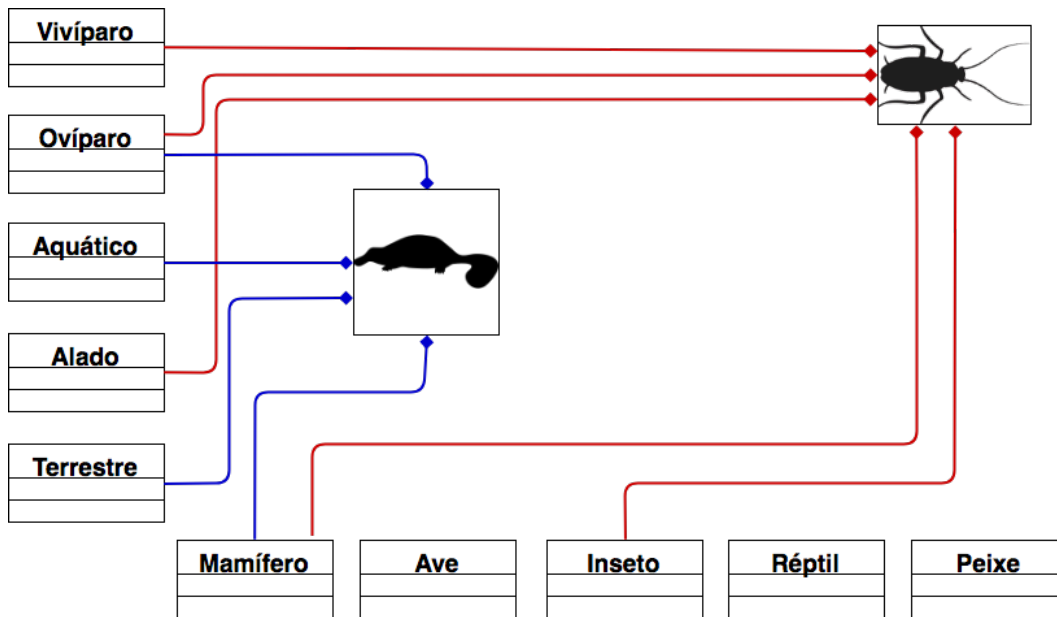


Figura 10 - Modelagem do reino animal utilizando ECS

No exemplo da Figura 10 podemos perceber que não há dificuldades em se modelar as características de qualquer animal, por mais fora de padrão que este seja, como o caso da Barata-besouro do Pacífico (*Diploptera punctata*) (ROTH; WILLIS, 1956), que é um inseto que produz leite.

É comum aos ECSs a definição de um *system-loop* ou *system-cycle*, onde cada passo de um determinado processo é executado em sequência. Quando no sistema existe o *system-loop* esses passos são executados ininterruptamente e, ao final do último passo, o processo executa o primeiro passo novamente, enquanto no *system-cycle* os passos de um determinado processo são executados apenas uma vez para cada recebimento de mensagem ou comando de execução.

Na Figura 11 é apresentado o modelo base de um ECS. Durante a execução o sistema (*System*) executa em ordem pré-estabelecida um método padrão (no exemplo, o método *run*) de cada entidade (*Entity*) que por sua vez executa um método padrão de cada componente. As interações de cada execução com os demais elementos do sistema são de responsabilidade de seu composto (chamador). Desta forma, a responsabilidade de criação (instanciação) e do gerenciamento das interações entre componentes é da entidade, assim como a responsabilidade de interações entre entidades é do sistema. Aos próprios objetos reside a responsabilidade de atuar sobre si mesmos.

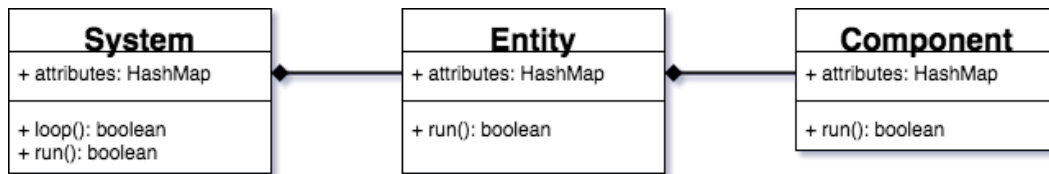


Figura 11 - Implementação base de um ECS

Neste ponto pode-se notar que um sistema, uma entidade e um componente possuem grande semelhança, porém suas responsabilidades são fortemente adversas. Wiebusch e Latoschik (2015) apresenta a responsabilidade de cada camada desta arquitetura conforme a seguir:

- O **sistema** é responsável pelo fluxo processual, coordena as entidades e as aciona no devido momento. É o gerente da organização.
- A **entidade** é responsável apenas pelas tarefas que lhe são atribuídas. Não conhece o processo da organização e nem possui capacidade de interferir neste fluxo. Possui um objetivo bem definido, sendo reativo e não intrusivo ao ambiente (sistema). É composto pelos componentes, que lhe atribuem todas as características.
- O **componente** é a unidade que possui os atributos descritores de uma determinada semântica e um comportamento relacionado, modificador de tais atributos, ativo ou reativo a mensagens da entidade a que pertence.

No trabalho de Hamza (2012) é apresentado o *framework* Artemis (disponível em <<http://gamadu.com/artemis/>>, acessado em :01-02-2018), construído utilizando a arquitetura ECS. Neste *framework* o componente possui apenas atributos, sem lógica complexa em sua implementação, seguindo o paradigma DOD (Data Oriented Design) (LLOPIS, 2009) e apresentado o ECS desta forma.

Para este trabalho seguiremos a implementação de Wiebusch e Latoschik (2015), onde os componentes possuem também comportamento ou funcionalidades.

Outro motivador para a utilização da arquitetura *Entity-component-system* e não a arquitetura com a abordagem hierárquica tradicional tem relação com a sustentabilidade de código (modularidade, modificação, reutilização e usabilidade da Interface de programação da aplicação) e também quanto a performance do

sistema. O trabalho apresentado na seção 3.3 apresenta essas questões, mostrando que, cada componente é completamente independente dos demais em sua implementação, uma entidade possui a capacidade de inserir e excluir componentes dinamicamente, assim como o sistema pode fazer o mesmo com as entidades (apenas respeitando os protocolos e interfaces pré-determinados). Esse total desacoplamento de código beneficia fortemente sua sustentabilidade (FISCHBACH; WIEBUSCH; LATOSCHIK, 2017).

2.6

Framework

Framework (ou arcabouço) possui diversas definições na literatura. Abaixo alguns autores que definiram o conceito:

- a) Fayad, Piatetsky-Shapiro & Smyth (1999b) e Johnson & Foote (1988) definem *framework* como um conjunto de classes que constitui um projeto abstrato para a solução de uma família de problemas.
- b) Mattsson (1996) define como uma arquitetura desenvolvida com o objetivo de atingir a máxima reutilização, representada como um conjunto de classes abstratas e concretas, com grande potencial de especialização.
- c) Johnson & Russo (1991) e Gamma e outros (1995) definem como um conjunto de objetos que colaboram com o objetivo de atender a um conjunto de responsabilidades para uma aplicação específica ou um domínio de aplicação.
- d) Buschmann et al. (1996) e Pree (1995) definem como um software parcialmente completo projetado para ser instanciado.

Apesar de diferentes, as definições encontradas na literatura não são contraditórias. O *framework* define uma arquitetura para uma família de subsistemas e oferece os construtores básicos para criá-los, além dos recursos e estruturas imutáveis do sistema para qualquer instanciação onde este seja aplicado, chamados de *frozen-spots*. Também são explicitados os lugares ou pontos de extensão, chamados de *hot-spots*, nos quais certos módulos ou funções devem ser implementados.

2.7

Arquitetura de Software

Arquitetura de software é um termo bastante amplo e abstrato, associado à definição das propriedades dos componentes de um sistema e suas relações. A seguir algumas definições (em tradução livre do autor):

Segundo Bass e outros, (2003, p. 14) *“A arquitetura de um sistema computacional são as estruturas do sistema, que compreende elementos de software, as propriedades visíveis e externas desses elementos, e as relações entre eles.”*

Já Fowler (2002, p. 27) diz que arquitetura é *“um termo que muitas pessoas tentam definir com pouca concordância, mas que há dois elementos comuns: um como o mais alto nível de abstração e componentização de um sistema; o outro como as decisões de construção difíceis de serem mudadas.”*

Krafzig e outros, (2004, p. 56) define arquitetura de software como *“Um conjunto de declarações, que descreve os componentes de software e atribui funcionalidades de sistema para cada um deles. Ela descreve a estrutura técnica, limitações e características dos componentes, bem como as interfaces entre eles. A arquitetura é o esqueleto do sistema e, por isso, torna-se o plano de mais alto-nível da construção de cada novo sistema.”*

Ainda segundo Krafzig e outros, (2004) uma das maiores funções de uma arquitetura de software é aumentar a agilidade e eficiência na construção e manutenção de sistemas, para isso, tal artefato deve possuir algumas características. São Elas:

- **Simplicidade:** A arquitetura deve ser simples, para permitir comunicação eficiente entre todas as partes interessadas e pessoas-chave do projeto de desenvolvimento e manutenção do sistema.
- **Flexibilidade e Manutenibilidade:** Todo sistema está sujeito à futuras evoluções. Precisa ser continuamente adaptado à novos requisitos. A arquitetura deve definir componentes distintos que pode ser reorganizado e reconfigurado de forma flexível. Mudanças locais não podem impactar o sistema globalmente.

- **Reusabilidade:** A reusabilidade tem sido um dos principais objetivos da engenharia de software. É de interesse do projeto que existam tantos “ativos de software” quanto possível. Isso pode ser conseguido criando um inventário de blocos de código úteis e reutilizá-los continuamente. O principal motivo para reutilização é a redução de custo de desenvolvimento e manutenção, o que pode ser conquistado compartilhando funcionalidades comuns em bibliotecas de códigos para uso em diferentes projetos.
- **Desacoplamento de Funcionalidade e Tecnologia:** A arquitetura não deve ser dependente de tecnologia e continuar válida através dos ciclos de inovação tecnológica, mantendo-se válida não só através dos ciclos de mudança na tecnologia de desenvolvimento de software como também através dos ciclo de vida reais dos artefatos tecnológicos instalados. Além disso, o desenvolvimento das funcionalidades deve ser desacoplado da tecnologia subjacente. A arquitetura deve evitar dependências à produtos e fornecedores específicos.

Estas características são fortemente aderentes às motivações explicitadas na seção 1.1 deste documento e a razão de a Arquitetura desenvolvida ser o principal artefato resultante deste trabalho.

3 Trabalhos Relacionados

Este capítulo apresenta os trabalhos existentes na literatura no que diz respeito a abordagens para projeto ou implementação de sistemas que se utilizam de técnicas de Aprendizagem de Máquina ou Visão Computacional para realizar a tarefa de identificar ou classificar anomalias em exames por imagem. Apresenta também um trabalho de construção de um sistema que utiliza a arquitetura *Entity-component-system* (ECS).

A descrição dos trabalhos relacionados será feita em diferentes tópicos.

No tópico 3.1 – Sistema de Visão Computacional para Detecção e Quantificação de Enfisema Pulmonar, (FÉLIX, 2007), é apresentado diferentes técnicas de segmentação de imagens e quantificação de atributos como área, volume e densidade dos segmentos encontrados. Apresenta também resultados que, apesar de não ser o estado da arte na matéria, são bastante promissores, dada a natureza generalista do trabalho e sua comparação com sistemas especialistas. O trabalho não realiza a tarefa de classificação de tais imagens segmentadas.

No tópico 3.2 – Desenvolvimento de um Sistema de Auxílio ao Diagnóstico de Pneumonia na Infância Utilizando Visão Computacional, (MACEDO et al., 2012), é apresentado o desenvolvimento de um sistema para auxiliar o diagnóstico de pneumonia nominado PneumoCAD, que avalia a probabilidade de o paciente estar com a doença, avaliando a sua radiografia através de classificadores previamente treinados com o intuito de auxiliar o médico na análise final e, como objetivo secundário, criar um mapa de possíveis focos da doença.

No tópico 3.3 – *Semantic Entity-Component State Management Techniques to Enhance Software Quality for Multimodal VR-Systems*, (FISCHBACH; WIEBUSCH; LATOSCHIK, 2017), é apresentado o desenvolvimento de um sistema de realidade virtual e realidade aumentada com interação multimodal que se utiliza da arquitetura ECS.

Ao final do trabalho o autor compara a abordagem hierárquica tradicional quanto a sustentabilidade de código (modularidade, modificação, reutilização e usabilidade da API) e também quanto a performance do sistema.

3.1

Sistema de Visão Computacional para Detecção e Quantificação de Enfisema Pulmonar, (FÉLIX, 2007)

Em (FÉLIX, 2007) o autor apresentou um protótipo de sistema de Visão Computacional implementado na linguagem Matlab para a segmentação automática dos pulmões, bem como para identificação e quantificação do enfisema pulmonar. Utiliza, para isto, um conjunto de imagens de Tomografia Computadorizada de voluntários. Realiza também uma comparação com um sistema de apoio ao diagnóstico de nome Osiris e os algoritmos SC (SILVA; CARVALHO, 2002) e SAB (SHOJAI; ALIREZAIE; BABYN, 2005). Estes algoritmos também são implementados em ambiente de simulação. O trabalho processa a imagem utilizando-se de quatro técnicas distintas, são elas: histograma, segmentação de imagens através de limiar, segmentação orientada a regiões e segmentação por transformada de *watershed*, (explicada a seguir).

O histograma, apresentado na Figura 12-B, fornece uma descrição global dos níveis de cinza da imagem. Já a segmentação de imagens através de limiar, tem como objetivo realçar conjuntos de pixels de determinada imagem que ocupam a mesma faixa de cores, correspondendo, normalmente, aos objetos de interesse e permitindo que seja efetuado um realce de um objeto em relação ao fundo, como exemplificado na Figura 12-C.

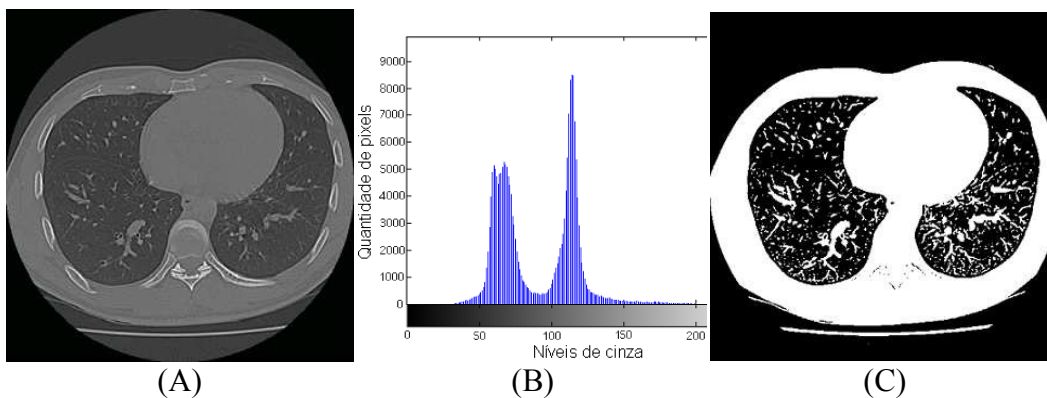


Figura 12 - Imagem de exame pulmonar (A), seu histograma (B) e limiarização (C) (FÉLIX, 2007)

A segmentação orientada a regiões realiza o agrupamento de subgrupos ou grupos de pixels em uma região desejada. A agregação das regiões é determinada pela escolha de uma semente na região em análise como, por exemplo, um pixel que pertença a esta região e que tenha direção e sentido definidos e características semelhantes às do objeto desejado. Estas características são normalmente obtidas por parâmetros de intensidade, média do local, variância local, entre outros (JAN, 2006).

A segmentação por transformada de *watershed* (BLEAU; LEON, 2000) é baseado na visualização de uma imagem 3D onde as coordenadas de uma imagem $g(x, y)$ e a intensidade do nível de cinza são interpretados como uma imagem topográfica, conforme demonstrado na Figura 13-B.

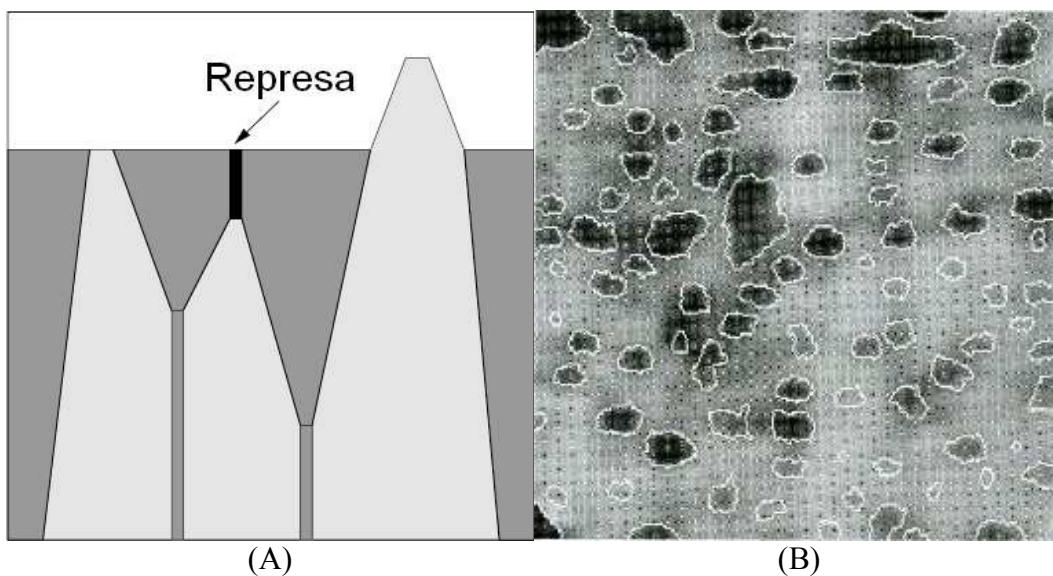


Figura 13- Perfil de inundação das bacias de captação com represa (A) e exemplo de segmentação (B) através de transformada de *watershed* (FÉLIX, 2007)

São considerados três pontos para sua interpretação: (i) os pontos (pixels) que pertencem a um mínimo regional; (ii) os pontos em que uma gota de água, se colocado no local de quaisquer desses pontos, cairia com certeza para um único mínimo, chamado de represa, conforme exemplificado na Figura 13-A; e, por último (ii) os pontos de divisores de água (*watershed*) nos quais é igualmente provável que a água caia, mais não se consegue determinar qual o seu mínimo (GONZALEZ; WOODS, 2002).

As estruturas do domínio específico (aparelho pulmonar, conforme exemplo na Figura 14) são segmentadas e analisadas através da combinação dessas técnicas e outras como:

- a) Fatiamento por densidade (GONZALEZ; WOODS, 2002),
- b) Segmentação por pseudo-cores (GONZALEZ; WOODS, 2002),
- c) Análise de medidas (perímetro, área, volume e densidade) (HARALICK, 1972; PRATT, 2001) e (ARAÚJO, 2004).

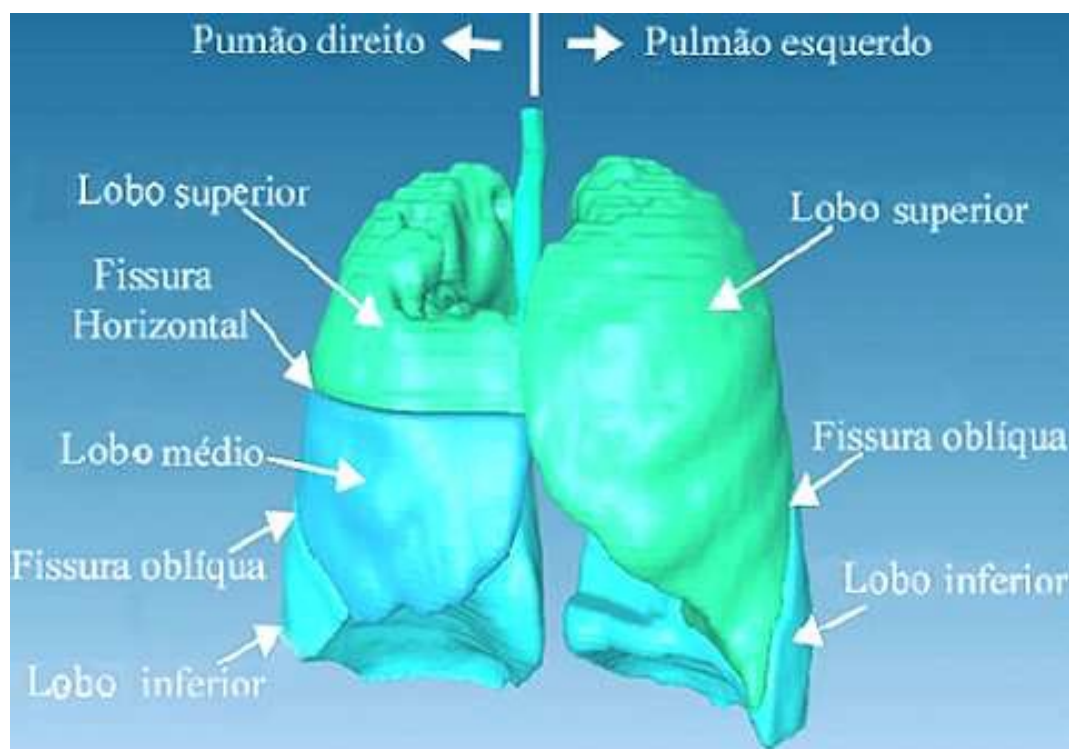


Figura 14 - Estruturas do aparelho pulmonar (FÉLIX, 2007)

A segmentação da imagem foi executada através de quatro algoritmos diferentes para avaliação e comparação. Os resultados das segmentações são classificados em:

- a) segmentação correta - ocorre quando todo o contorno do tecido pulmonar é segmentado sem perdas ou acréscimo de estruturas;
- b) sobre-segmentação - acréscimo de estruturas que não fazem parte do tecido;
- c) segmentação com perdas - perda de parte de tecido relevante para análise;
- d) segmentação errada - quando o tecido não é segmentado.

Na Figura 15 é apresentada uma avaliação de corretude na segmentação do aparelho pulmonar utilizando cada um dos algoritmos executados.

	Imagens	Segmentação correta	Sobre-segmentação	Segmentação com perdas	Segmentação errada
Osiris	voluntário	61,76%	0,00%	38,24%	0,00%
	paciente	52,48%	0,71%	46,81%	0,00%
SC	voluntário	48,04%	46,08%	1,96%	3,92%
	paciente	67,38%	24,11%	1,42%	7,09%
SAB	voluntário	68,63%	28,43%	1,96%	0,98%
	paciente	70,21%	21,98%	4,26%	3,55%
SDEP	voluntário	94,12%	4,90%	0,98%	0,00%
	paciente	90,78%	4,26%	4,96%	0,00%

Figura 15 - Avaliação da segmentação para cada algoritmo estudado (FÉLIX, 2007)

Após a segmentação o trabalho propõe o uso de histograma para quantificação da evolução da Doença Pulmonar Obstrutiva Crônica (DPOC).

As funções usadas para segmentação de imagem, em seu mais alto nível, possuem uma interface com assinatura $f(I) = I'$, recebe uma imagem colorida ou em tons de cinza (I) e retorna uma imagem em duas cores (*Bitmap*) com o mapa da segmentação (I'). Um exemplo de entrada e saída desse processo pode ser visto na Figura 12-A e na Figura 12-C, nesse exemplo $f(A) = C$. A arquitetura proposta por essa dissertação, por definição de seus requisitos, deve ser compatível com este tipo de função.

3.2

Desenvolvimento de um Sistema de Auxílio ao Diagnóstico de Pneumonia na Infância Utilizando Visão Computacional, (MACEDO et al., 2012)

Em (MACEDO et al., 2012) é apresentado o desenvolvimento de um sistema para auxiliar o diagnóstico de pneumonia nominado PneumoCAD.

O PneumoCAD avalia a probabilidade de o paciente estar com a doença por meio de sua radiografia, avaliando-a através de classificadores previamente treinados. O resultado é apresentado ao médico para auxiliá-lo na análise final. Como objetivo secundário, o sistema marca em um mapa o endereço da casa e/ou creche do paciente para que seja possível identificar a geolocalização de possíveis focos da doença.

Foram utilizadas técnicas de reconhecimento de padrões em texturas por meio da decomposição das transformadas *wavelets*, das características extraídas das decomposições e da classificação aplicadas as radiografias. Em duas metodologias investigou-se qual melhor característica aplica-se ao reconhecimento de padrões em texturas e qual o melhor desempenho do método classificador K-NN. O procedimento iniciou-se com a aplicação da transformada wavelet de Haar e com a extração de 17 características de cada radiografia que foram armazenadas em descritores. A metodologia I testou o aumento da acurácia da classificação, proporcionalmente ao aumento balanceado da quantidade de radiografias de cada classe. A metodologia II testou a capacidade do K-NN de gerar classificações em níveis aceitáveis com o desbalanceamento aleatório da quantidade de imagens entre as duas classes. Obteve-se um nível médio de acurácia de 91,75% com destaque para a característica diferença de variância e o desempenho do método K-NN mostrou-se mais eficaz quando a quantidade de vizinhos mais próximos foi de $K=9$. Os resultados são considerados promissores, concluindo que o sistema utilizado (PneumoCAD) pode ser uma ferramenta útil no diagnóstico em pneumonia na infância, aliando o conhecimento homem-máquina e oferecendo condições para a

interpretação de radiografias torácicas em "padrão-ouro", conforme especificações da OMS. Além disso, o software se apresenta como uma tecnologia inovadora em saúde, ao disponibilizar para administradores e gestores políticos uma ferramenta de controle e monitoramento epidemiológico da pneumonia em tempo real, produzindo subsídios para as decisões e organizações dos serviços de saúde relacionados.

Desenvolvido com recursos da Fundação de Amparo à Pesquisa do Estado de Goiás (Fapeg), em parceria com o Instituto de Patologia Tropical Saúde Pública (IPTSP) da UFG, em 2016 tal projeto se encontrava em funcionamento de forma experimental no Instituto de Informática da Universidade Federal de Goiás (INF-UGF), ainda não foi colocado em nenhuma instituição de saúde. O pesquisador afirma que pretende firmar parcerias com o Governo Estadual ou com a Prefeitura de Goiânia para disponibilizar o sistema em um projeto-piloto.

Este trabalho utiliza o termo visão computacional para descrever classificadores de imagem sem se utilizar de técnicas de segmentação de imagem para melhorar a acurácia da classificação.

Diversos outros trabalhos, como aqueles citados na seção 1.3 deste documento, apresentam a mesma abordagem para resolver o problema de identificação de doença

A geração de classificadores através de aprendizado supervisionado pressupõe que exista uma base de dados de exames devidamente classificados quem possuam a doença em estudo (pneumonia), outras doenças e imagens saudáveis em quantidade suficiente para que sejam gerados modelos de classificação eficientes.

As funções usadas para classificação de uma nova imagem, em seu mais alto nível, possuem uma interface com assinatura $f(I, M) = L$, recebe uma imagem colorida, em tons de cinza, ou duas cores (*Bitmap*) (I), um modelo de classificação de imagens (M) compatível com a imagem (I) e retorna um rótulo de classe (*Label*) (L) com a classe mais provável a que (I) pertence. A arquitetura proposta por essa dissertação, por definição de seus requisitos, deve ser compatível com este tipo de função.

3.3

Semantic Entity-Component State Management Techniques to Enhance Software Quality for Multimodal VR-Systems, (FISCHBACH; WIEBUSCH; LATOSCHIK, 2017)

Em (FISCHBACH; WIEBUSCH; LATOSCHIK, 2017) o autor apresenta o padrão *Entity-component-system* (ECS) (ALATALO, 2011) como uma abordagem proeminente para o dilema de acoplamento, como apresentado por Lange (2014) e Latoschik (2011). Esse padrão organiza os dados (componentes) associados aos subsistemas em uma visão centrada no objeto (as entidades) usando a composição ao invés de herança. Esta composição melhora o desacoplamento, resolvendo problemas que ocorrem em casos onde os subsistemas precisam de acesso mútuo a componentes fora de sua associação primária de dados. São exemplos típicos os subsistemas de Inteligência Artificial que incorporam modelos de dados que refletem o estado geral da aplicação, para fornecer capacidades de inferência ou aterramento semântico e subsistemas para interfaces multimodais (MMI) como técnicas de interação alternativas promissoras para sistemas interativos em tempo real (Real-time interactive systems, ou RIS) (LATOSCHIK, 2005; AMERI EKHTIARABADI et al., 2011; CHERUBINI et al., 2015)

Neste trabalho Fishbach se utiliza da arquitetura ECS para construção de uma interface multimodal com semântica para sistemas interativos em tempo real (RIS) como Realidade Virtual, Realidade Aumentada e Mista (VR, AR e MR), jogos de computador e robótica. Um exemplo de interação em uma interface multimodal é um usuário apontar para um objeto em uma interface gráfica, falar “coloque isto ali”, apontando para os pontos de origem e destino do objeto desenhado e o sistema reagir como desejado. A Figura 16 apresenta algumas dessas interfaces construídas durante o trabalho de Macedo e outros (2012).



Figura 16 - Exemplos de interfaces multimodais (MACEDO et al., 2012)

Para executar tal tarefa as unidades das estruturas da MMI precisam de acesso, em tempo real, a informações que vão desde a entrada do sensor bruto no nível de dados até a informação semântica de um determinado contexto no nível de decisão. Ao mesmo tempo, eles devem ser desacoplados do circuito de simulação principal para não comprometer a simulação geral. Por isso, o aumento dos requisitos de acesso e desacoplamento do estado da combinação MMI-RIS são particularmente adequados como casos de uso para o desenvolvimento de soluções de melhor qualidade de software de arquiteturas RIS complexas (LATOSCHIK, 2005 e FISCHBACH, 2015). Para reagir às interações, cada objeto (ou entidade) possui um conjunto de ações possíveis, definido por seus atributos e funções (componentes).

Como exemplo, se tomarmos um quadrado e um cubo, poderíamos comandar para ambos as ações de *mover*, *alterar tamanho* e *rotacionar*, mas apenas o cubo poderia *mostrar a face à direita*.

Aproveitando-se da arquitetura ECS o sistema pode dinamicamente adicionar comportamentos a partir da verificação de atributos existentes. Um objeto só pode rotacionar no eixo Z se possuir 3 dimensões, somente pode sofrer a ação da gravidade se possuir massa, etc.

O artigo ainda apresenta cinco técnicas baseadas em semântica que estendem a arquitetura ECS e visam explicitamente manter a sustentabilidade, isto é, modularidade, modificação, reutilização e usabilidade da API (PICCIONI; FURIA; MEYER, 2013) para sistemas VR, AR e MR. As técnicas fornecem um esquema de acesso unificado ao estado da aplicação e facilitam a integração de métodos de IA simbólicos, ilustrado para o processamento multimodal de nível de decisão.

O trabalho conclui que a utilização de ECS aprimora as capacidades do padrão de representação e acesso ao estado da aplicação, introduzindo um acesso semântico uniforme em diferentes níveis e facilitando a criação de subconjuntos desacoplados. Elabora também um método de avaliação de características de desempenho, incluindo uma comparação com abordagens alternativas de representação do estado, validando a viabilidade das técnicas apresentadas. Os aspectos abordados independem de uma determinada linguagem de programação, o que permite aos desenvolvedores se beneficiar das ideias apresentadas e implementá-las em seus próprios softwares.

Uma implementação de referência é exibida por várias amostras de código, que são os resultados de um processo de revisão de pares de API conduzidos para melhorar a usabilidade da API. Seu código fonte está disponível em <<http://www.hci.uni-wuerzburg.de/projects/simulator-x.html>> (acessado em 06-2017) para a comunidade de pesquisa para coletar feedback e promover a reutilização.

Este trabalho apresenta um exemplo de sistema de computação interativa, conforme abordado por Zaks (2017) e apresentado na seção 2.5 deste documento, onde pode-se observar a simplificação da arquitetura do sistema em sua construção ao se fazer uso da ECS para se separar estado e comportamento dos elementos, adicionando-os na etapa de construção do objetos ao invés de fazê-lo na etapa de definição de classes. O trabalho apresentado nessa dissertação aplica de forma semelhante a arquitetura ECS para resolução do problema proposto.

4 Solução Proposta

A solução proposta contempla entre outros artefatos um *framework* para a criação do sistema configurável no domínio especificado. Tal sistema utiliza o padrão de arquitetura de software denominado *Entity-component-system* descrito na seção 2.4 e exemplificado na seção 3.3 deste documento.

Foi definido também um processo para a segmentação e classificação de um novo exame, assim como a identificação de áreas na imagem que possuam alta probabilidade de possuir marcadores patológicos. Baseado no KDD, o novo processo apresenta as seguintes alterações em relação as fases originalmente definidas:

- **Inclusão da etapa de segmentação;** apesar de fundamentalmente ser uma etapa de transformação de dados, esta etapa se distancia do conceito definido como transformação no KDD, onde o objetivo da etapa é tratar dados para facilitar a fase de mineração, conforme mencionado da seção 2.1 deste documento. No caso da segmentação, a fase possui o objetivo explícito de incluir processos de segmentação de imagem, gerando novos dados a partir da imagem original, visando aprimorar o trabalho das fases de extração de atributos de domínio e classificação.
- **Inclusão da etapa Extração de Atributos de Domínio;** tem sua fundamentação nas etapas de transformação e interpretação de dados do KDD. A responsabilidade dessa fase é interpretar as informações obtidas na fase de segmentação de acordo com algoritmos que representem conhecimento do domínio para posterior comparação com valores normais contidos na base de conhecimento.

Essas duas fases são descritas em detalhes na seção 4.1 deste documento.

Uma das grandes vantagens da arquitetura desenvolvida é a possibilidade de alterar e incluir comportamentos e atributos ao sistema em tempo de execução,

possibilitando que o sistema se configure para atender as demandas específicas de uma necessidade processual sem parada ou recompilação do mesmo.

Na seção 4.1 será apresentado o processo proposto, necessário para a execução da solução.

Na seção 4.2 será apresentado o Diagrama de Classes do Modelo, apresentando as classes detentoras dos dados que irão trafegar pelo sistema.

Na seção 4.3 será apresentada a implementação *Entity-component-system* utilizada e seu ciclo de execução.

4.1

O Processo

O processo proposto por este trabalho é baseado no KDD (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996) e fortemente influenciado pelo sistema típico de processamento de visão computacional apresentado por Filho & Neto (1999). Foi adicionado ao processo etapas para a execução das tarefas de classificação de imagens e interpretação dos dados gerados em informação útil ao médico. A seguir será apresentada cada etapa deste processo, assim como os recursos e artefatos que compõem e disparam o processo do interpretador. A Figura 17 apresenta uma visão geral do fluxo de processamento para interpretação de exame.

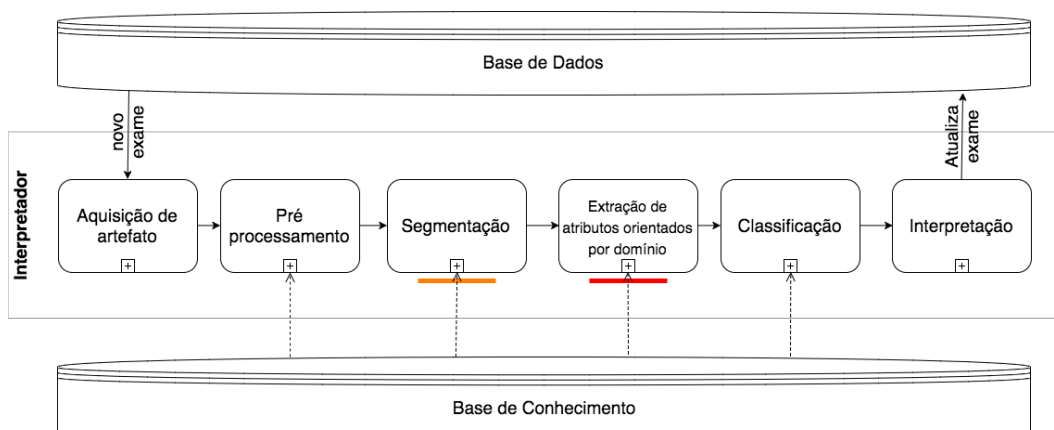


Figura 17 - Fluxo de processo para interpretação de exame

4.1.1

Novo Exame

Artefato gerado e persistido na base de dados em conformidade com um modelo de dados compatível com o processo Aquisição de Artefato (que será apresentado a posteriori). O processo de geração de um novo exame não é objeto de pesquisa deste trabalho, necessita-se apenas saber que existe um artefato gerador de exames por imagem que interage com um usuário que, após a geração da imagem de exame, gera um conjunto de metadados. Um conjunto de imagens mais os metadados compõem um novo exame. O novo exame é salvo na base de dados em conformidade com um modelo de dados pré-estabelecido, adicionando-se também o status ‘novo’ à este exame.

Baseado na etapa 1 (coleta de dados) do KDD apresentado na seção 2.1 deste documento.

4.1.2

Aquisição de Artefato

Responsável por receber a informação que existe um novo artefato a ser processado e convertê-lo em um objeto do sistema para processamento.

Baseado na etapa 2 (seleção) do KDD apresentado na seção 2.1 deste documento e adaptado. Neste caso todos os dados de um determinado exame são recuperados, cada componente do sistema será responsável por definir que elementos do exame serão processados por ele, adicionando ao exame os resultados deste processamento.

O passo-a-passo desta etapa é a seguinte:

1. Adicionar um observador à base de dados, que avisa se há ou não novos exames para processamento (exames com status ‘novo’).
2. Recuperar o exame informado pelo observador e transformá-lo em um objeto da classe *exam* (que será apresentada na seção 4.2 deste documento), mantendo-o em memória e acessível.
3. Alterar o status do exame na base de dados para ‘em processamento’.
4. Enviar mensagem à entidade responsável pelo próximo passo para dar segmento ao processamento.

4.1.3

Pré-processamento

Responsável por receber um artefato do tipo exame e padronizá-lo para utilização dos processos seguintes. O pré-processador necessita saber o tipo de exame que está sendo processado e o padrão adotado pelo sistema para este exame (dados que podem ser adquiridos a partir de uma consulta à base de dados ou à base de conhecimento), transformando seus dados para que este se torne comparável com os demais exames na base de dados e compatível com as informações da base de conhecimento. Alguns exemplos de processos nesta etapa são redimensionamento de imagem, correção de gama de luz e cores de uma imagem e transformação de dados numéricos de valores absolutos para percentuais, assim como identificação do tipo de exame recebido e verificação de consistência dos dados do exame.

Baseado na etapa 3 (pré-processamento) do KDD apresentado na seção 2.1 deste documento.

O passo-a-passo desta etapa é a seguinte:

1. Acessar o exame em memória.
2. Verificar que tipo de exame será processado.
3. Recuperar da base de conhecimento quais transformações são necessárias aos exames deste tipo para que o processamento possa prosseguir.
4. Executar as transformações de dados e processamentos necessários, atualizando o exame em memória.
5. Enviar mensagem à entidade responsável pelo próximo passo para dar segmento ao processamento.

4.1.4

Segmentação

É nesta etapa que se aplicam os algoritmos de visão computacional nos artefatos do tipo imagem. Cada imagem de entrada gera um conjunto de artefatos de saída, cada artefato corresponde à uma subestrutura anatômica apresentada na imagem original. Dependendo do algoritmo de segmentação utilizado, pode ser

necessário a recuperação de conhecimento à partir de uma base de conhecimento, seja ela alguns parâmetros como os utilizados em algoritmos como o *Multiresolution Segmentation* (BAATZ; SCHÄPE, 2000) onde pode-se definir alguns poucos parâmetros como escala e homogeneidade, associando parâmetros ideais a cada tipo de exame, seja ele mapas (atlas) de segmentação, como os utilizados em algoritmos como o *Active Appearance Model* (AAM) (EDWARDS; COOTES; TAYLOR, 1998), onde se utiliza um Atlas probabilístico da região onde uma determinada estrutura acontece e o algoritmo executa um ajuste para que a segmentação se adeque à realidade da amostra. A utilização de algoritmos que usam de atlas para a segmentação é interessante na segmentação de imagens de exames médicos pois abre a possibilidade de se executar a segmentação e rotular cada segmento com o nome da subestrutura segmentada em uma mesma etapa de processo.

Para melhor entendimento, tomemos uma imagem de exame (I) que possua diversas subestruturas ($S_1...S_n$). Para sua segmentação, o sistema deve possuir uma entidade (E) capaz de executar tal tarefa. Esta entidade deve possuir um componente (C) que implementa um algoritmo de segmentação (A), configurado para extrair da imagem uma subestrutura (S) para cada uma das subestruturas ($S_1...S_n$).

Na Figura 18 é apresentada uma ilustração da segmentação automatizada do tecido lombar: (a) mostra a MRI original do meio sagital, (b) e (c) mostram a segmentação manual (conhecimento base para aprendizado, que pode ser usado como atlas), (d), (e) e (f) mostram os mapas dos rótulos para o saco dural, disco e vértebra, respectivamente, usando os métodos de mapa de probabilidade e características de HOG (*Histogram of Oriented Gradients*) (OREIFEJ; LIU, 2013). Neste exemplo, podemos definir a subestrutura anatômica (S_1) como a medula espinhal, representado em vermelho na Figura 18-a, (S_2) como os discos intervertebrais, representado em verde na Figura 18-b, e (S_3) como as vértebras, representado em azul na Figura 18-c. Para que a tarefa seja executada, o sistema deve ter uma entidade (E) que saiba processar uma ressonância magnética de coluna lombar, e os componentes (C_1), (C_2) e (C_3) que consigam segmentar respectivamente (S_1), (S_2) e (S_3) da imagem original. Perceba que o grau de reusabilidade de cada um dos componentes (C_1), (C_2) e (C_3) é tão alto quanto são as subestruturas anatômicas compatíveis com seu processo de segmentação.

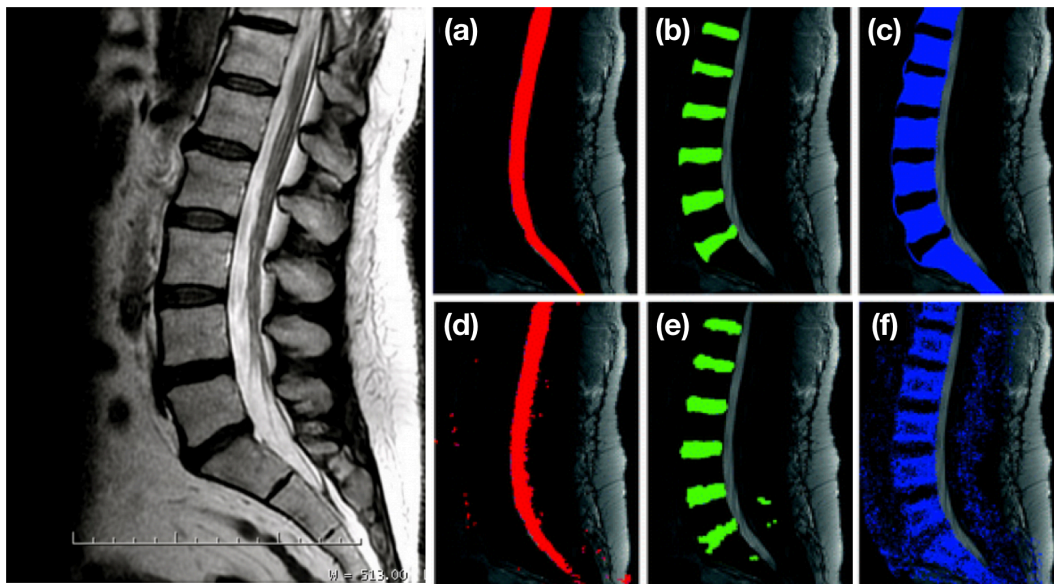


Figura 18 - Exemplo de métodos de segmentação de uma imagem de ressonância magnética da coluna lombar

Esta etapa, apesar de conceitualmente poder ser considerada uma etapa de transformação de dados (etapa 4 do KDD) seu grau de complexidade o distancia dos processos usuais de transformação comumente descritos. Em Dias e outros. (2012) a etapa de transformação do KDD é descrita como os processos que armazenam os dados e os agrupam adequadamente em um formato apropriado para a mineração. Também consiste no pré-processamento final dos dados, removendo os ruídos, tratando os atributos ausentes, padronizando os atributos e removendo os registros repetidos. A etapa sugerida cria novas informações através da adição de conhecimento externo aos dados originais e realização de complexo processamento dos dados para geração de novos dados, expandindo em muito o conceito original da definição desta fase do processo KDD. O autor, portanto, considera esta fase como uma adição ao processo base.

O passo-a-passo desta etapa é a seguinte:

1. Acessar o exame em memória.
2. Verificar que tipo de exame está sendo processado.
3. Recuperar da base de conhecimento quais segmentações são necessárias aos exames deste tipo para que o processamento possa

prosseguir, assim como os artefatos necessários para execução de tais segmentações.

4. Executas as segmentações e processamentos necessários, adicionando ao exame em memória as informações das segmentações realizadas, como parte das informações do exame.
5. Enviar mensagem à entidade responsável pelo próximo passo para dar continuidade ao processamento.

4.1.5

Extração de Atributos Orientados por Domínio

Nesta etapa cada artefato obtido na etapa de segmentação é submetido a algoritmos especializados, construídos com o objetivo de avaliar características selecionadas a partir de conhecimento do domínio. O objetivo desta etapa é transportar o conhecimento já existente no domínio para dentro do sistema através processos de instrução ao invés de experiência.

Para clarificar a diferença entre geração de conhecimento através de processos de instrução e de conhecimento, podemos citar como exemplo, a mensuração ultrassonográfica da espessura das camadas íntima e média das artérias carótidas (EIMC ou IMC) (ROELKE et al., 2013). Na Figura 19 é apresentado um exemplo de medida da espessura do complexo médio-intimal (IMT) na carótida primitiva, o qual é indicado através dos traços horizontais marcados na parede posterior do vaso. Os trabalhos de Roelke e outros (2013) e Júnior e outros, (2013) concluem que esta medida é importante como marcador precoce de aterosclerose e para avaliação do risco de eventos cardiovasculares. O ultrassom, utilizado na sua avaliação, é um exame acessível e de baixo custo (DATASUS, 2015). São descritas medidas realizadas em diferentes regiões das carótidas. Os valores de normalidade da medida são claramente apresentados no trabalho.

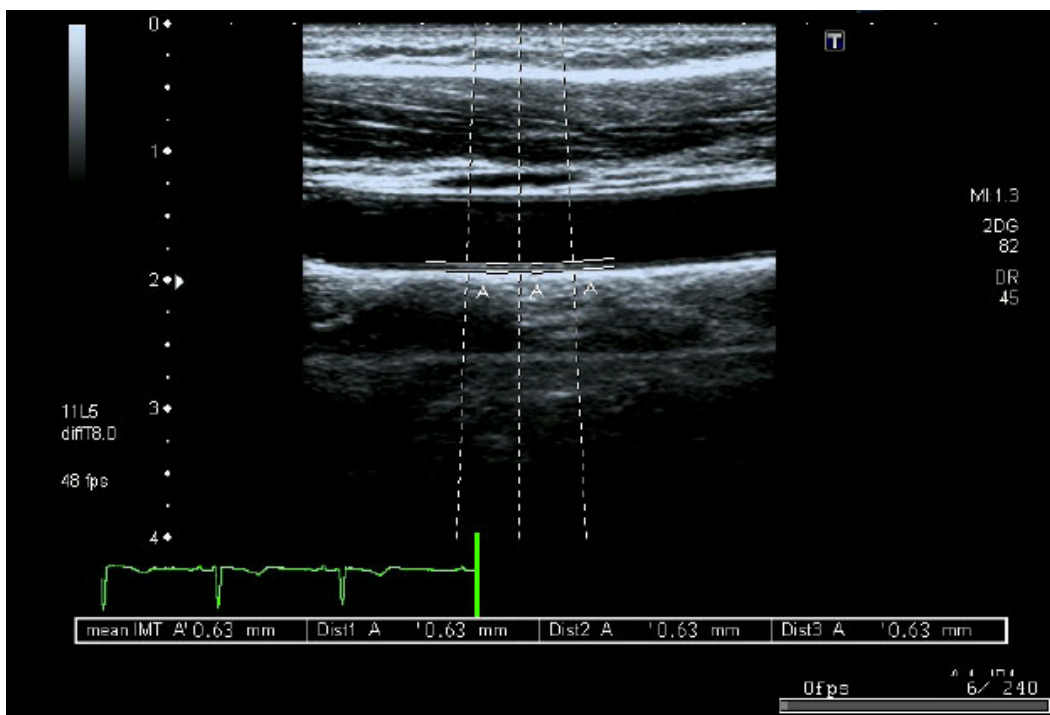


Figura 19 - Ultrassonografia da carótida com EIMC

Utilizando este caso compararemos as duas abordagens possíveis:

- **Deteccção através de processos de instrução:**

Neste caso um algoritmo especializado em verificar a espessura da parede da carótida na imagem é gerado e o valor obtido comparado com os valores preestabelecidos. Desta comparação é obtida a classificação da instância, podendo esta ser n-ária ou escalar.

- **Deteccção através de aprendizado:**

Como apresentado na seção 2.2 deste documento, neste caso é necessário a obtenção de uma coleção de exames de mesmo tipo previamente classificados e um algoritmo de aprendizagem de máquina para a geração de um modelo para classificação. A instância à ser pesquisada é comparada com o modelo para geração de uma classificação n-ária.

A inserção desta etapa se fez necessária após a análise de experimentos executados pelos autores indicarem que para alguns casos o trabalho realizado para geração de um algoritmo para deteção através de processo de instrução ser significativamente menor do que o trabalho realizado para geração do modelo

classificador através de aprendizado de máquina. Também em alguns casos obteve-se resultados melhores com esta primeira abordagem, além de a possibilidade de resultados escalares serem particularmente interessantes para o objetivo desejado descrito no tópico (b) da seção 7.2 deste documento.

Apesar desta etapa ser baseada nas etapas 4 e 6 do KDD (transformação e interpretação) apresentado na seção 2.1 deste documento, foi adicionado o conceito de inserção do conhecimento do domínio neste processo, onde se aplicam algoritmos que possibilitam a extração de informações específicas para verificar alguma característica apontada como importante por orientação especializada no domínio e para o domínio.

O passo-a-passo desta etapa é a seguinte:

1. Acessar o exame em memória.
2. Verificar que tipo de exame está sendo processado.
3. Recuperar da base de conhecimento quais atributos de domínio são relevantes para reconhecimento.
4. Verificar se existem partes do exame compatíveis com realização do processamento necessário para a aquisição dos atributos de interesse.
5. Realizar o processamento necessário para a aquisição dos atributos de interesse, adicionando ao exame em memória os valores destes atributos.
6. Enviar mensagem à entidade responsável pelo próximo passo para dar continuidade ao processamento.

4.1.6

Classificação

Nesta etapa será executada a classificação da imagem. Esta classificação é feita à partir funções de inferência que se utilizam de modelos criados por aprendizagem de máquina tendo como valor de saída uma categoria (classe) ou valor escalar. O tipo de saída dependerá diretamente da técnica utilizada, podendo ser um booleano, um rótulo ou um valor escalar.

A classificação poderá ocorrer de duas formas:

- **Classificação a partir de Modelo Preditor**

Neste caso o componente classificador se utiliza de um modelo gerado por aprendizagem de máquina (conforme apresentado na seção 2.2 deste documento) e uma função que, utilizando tal modelo, seja capaz de rotular o exame (ou parte dele) dentre um conjunto de rótulos (classes) conhecidos. Tal modelo deve estar presente na base de conhecimento do sistema.

- **Classificação por Detecção de Anomalias**

Neste caso o componente classificador se utiliza de um modelo ou matriz de covariância (conforme mencionado na seção 2.3 deste documento) e uma função que, utilizando tal artefato, seja capaz de avaliar o grau de normalidade em que o exame (ou parte dele) se encontra, retornando um escalar representando este valor. Tal modelo ou matriz de covariância deve estar presente na base de conhecimento do sistema.

Baseado na etapa 5 do KDD apresentado na seção 2.1 deste documento, esta etapa difere da etapa apresentada na seção 4.1.5 deste documento (Extração de Atributos Orientados por Domínio) pois os artefatos classificadores não possuem conhecimento explícito do domínio nem orientação direta para detecção de características de interesse. Os modelos aos quais a etapa é dependente são gerados fora do sistema. Este trabalho sugere que se amplie o reaproveitamento dos modelos gerados pelos trabalhos de pesquisa no campo de ML (aprendizagem de máquina) para apoio à medicina diagnóstica. Esta etapa permite que os diversos modelos com esta finalidade produzidos pela comunidade científica sejam utilizados e tenham seus acessos ampliados. Tal iniciativa já ocorre em outros domínios como por exemplo o *Apple ML Model Repository* (APPLE, 2018), onde pode-se também se verificar o interesse da comunidade na utilização dos modelos prontos.

O passo-a-passo desta etapa é a seguinte:

1. Acessar o exame em memória.
2. Verificar que tipo de exame está sendo processado.
3. Recuperar da base de conhecimento os modelos de classificação compatíveis com o exame.
4. Realizar o processamento necessário para a classificação do exame, adicionando ao exame em memória os valores e/ou rótulos destas classificações.

5. Enviar mensagem à entidade responsável pelo próximo passo para dar continuidade ao processamento.

4.1.7

Interpretação

Interpretação é o ato de fazer sentido (FERREIRA, 1996), transformar informação em conhecimento, e este é o objetivo desta etapa, transformar todas as informações geradas nas etapas anteriores em conhecimento e facilitar a apresentação destes resultados ao usuário. Exemplos de saídas para esta etapa podem ser: um marcador de área de uma imagem com grande probabilidade de possuir alguma anomalia; um quantificador percentual do grau de anomalia de um exame, para possível ordenação de seu grau de severidade em relação aos demais exames da base ou; um rótulo indicando a provável presença de alguma patologia.

Após o processo de interpretação, a informação resultado pode ser persistida na base de dados, e deve ser persistida para que seja possível comparar resultados entre diferentes exames de mesmo tipo.

Baseado na etapa 6 do KDD apresentado na seção 2.1 deste documento.

O passo-a-passo desta etapa é a seguinte:

1. Acessar o exame em memória.
2. Verificar que tipo de exame está sendo processado.
3. Recuperar da base de conhecimento as informações desejáveis à exposição para este tipo de exame.
4. Realizar o processamento necessário para a apresentação das informações relevantes e presentes no exame.
5. Persistir todas as novas informações na base de dados.
6. Alterar o status do exame para 'processado'.
7. Enviar mensagem à entidade responsável, informando o fim do processamento.

4.2

Diagrama de Classes do Modelo

Para se conseguir o objetivo esperado, há a necessidade de se definir uma estrutura de classes de objetos que serão operados pelas diversas partes do processo. Tais classes devem ser flexíveis o suficiente para acomodar os inúmeros exames passíveis de serem submetidos aos processos do sistema, e rígidos o suficiente para que tais processos não necessitem sofrer modificações para avaliar novos tipos de exames.

O modelo possui um único ponto de extensão (*hot-spot*), representada pela classe abstrata **ExamPartData** e oito classes fixas (*frozen-spots*) para representar as informações comuns à todos os exames.

Para atingir este fim, foi definido como parte desta arquitetura as classes e relações como mostrado na Figura 20 e descritas abaixo:

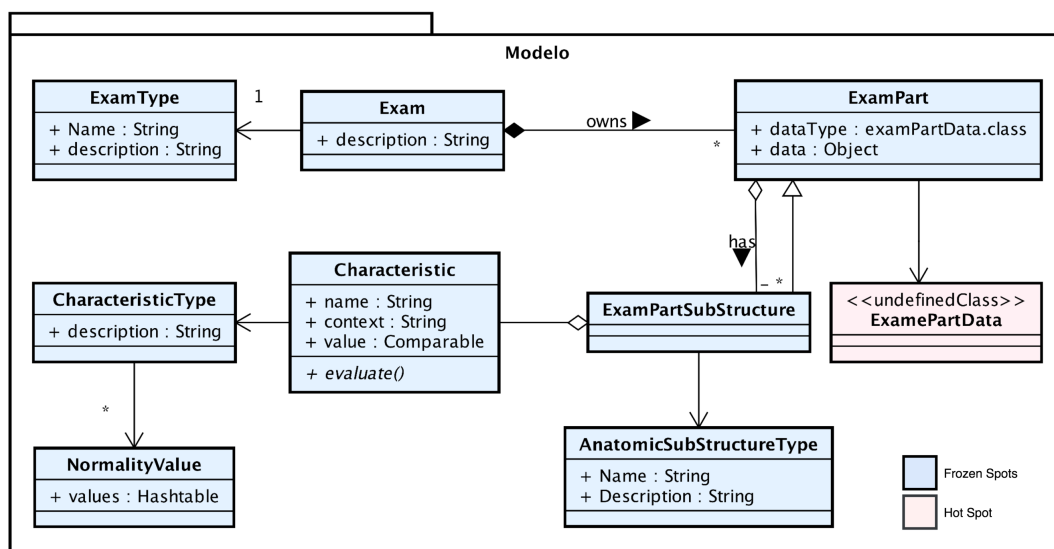


Figura 20 - Pacotes e classes da camada de modelo do sistema proposto

- Um Tipo de Exame (**ExamType**) categoriza os exames para facilitar o agrupamento semântico dos dados. Essa informação é usada no processo para ajudar a associar exames com entidades que os processam.

- Um Exame (**Exam**) possui um conjunto de partes. (*Ex: um exame de tomografia computadorizada possui um conjunto de imagens + dados adicionais sobre o exame*)
- Cada Parte de Exame (**ExamPart**) é uma unidade dado, definida por uma classe que a modela. Esta classe deve ser implementada para cada tipo de parte de exame tratado pelo sistema. A classe ExamPart guarda os dados e o tipo de classe a que esses dados pertencem, generalizando-o para o sistema sem perder o modelo original dos dados. Uma Parte de Exame possui um conjunto de Subestruturas não explicitamente reveladas, que podem através de processamento serem reveladas e apresentadas como novas informações. Por exemplo, uma imagem de raio-x de tórax possui um grande conjunto de subestruturas que podem ser identificadas e segmentadas através de algoritmos ou manualmente.
- Os Dados da Parte do Exame (**ExamPartData**) é a única estrutura de dados completamente abstrata e o único ponto de extensão (*hot spot*) desta estrutura de dados. Ela deve conter os dados de uma unidade de informação de um exame, como uma imagem fotográfica ou uma fatia de imagem em um exame tomográfico por exemplo. Esta classe deve suportar o formato deste dado, que esta sendo tratado como dado abstrato pela classe ExamPart, assim como possuir as funções necessárias para sua leitura. Ao se adicionar uma nova classe deste tipo no sistema, deve-se também adicionar um componente ao ECS que a suporte e processe.
- Uma Substrutura (**ExamPartSubStructure**) é gerada a partir do processamento de uma parte de exame através de um algoritmo de geração e pode ou não ser rotulada com uma subestrutura anatômica (**AnatomicSubStructureType**). Tomando a Figura 21 como exemplo, podemos segmentar a parte da imagem que contém os pulmões utilizando técnicas de visão computacional. Essa nova imagem possui características como por exemplo área, densidade e volume estimado.
- Uma característica (**Characteristic**) é relacionada com uma subestrutura, possui um tipo (**CharacteristicType**) os valores das características devem conter dados comparáveis com valores padrão.

- Um tipo de característica (**CharacteristicType**) rotula uma característica obtida e possui valores padrão de normalidade (**NormalityValue**)



Figura 21 - Radiografia de tórax normal em PA apontando as principais estruturas anatômicas identificáveis

As entidades e componentes que tenham por objetivo processar os dados médicos deverão fazê-lo utilizando este modelo. A motivação é não criar elementos com dependências entre componentes através de modelos externos à arquitetura, mantendo estável e controlado o nível de acoplamento do sistema.

4.3

Implementação Usando ECS

No ECS há apenas 3 tipos de classe à implementar. Para a arquitetura proposta temos que as classes *System* e *Entity* são inalteráveis (*frozen-spots*), instanciadas e configuradas na inicialização do sistema. A classe *Component* é o ponto flexível (*hot-spot*) desta parte do sistema, possuindo o método abstrato *run()* pendente de implementação. Cada processo especialista do sistema necessariamente é uma subclasse de *Component*, com o método *run()* implementado, representando uma etapa do processamento de um exame. Na Figura 22 temos as classes do ECS com seus *hot-spots* e *frozen-spots* identificados.

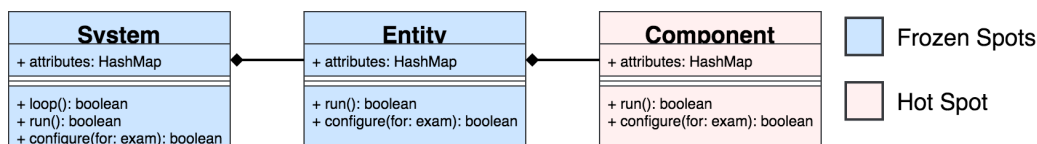


Figura 22 - ECS com identificação de *Hot-spots* e *Frozen-spots*

Foi adicionado às classes do ECS o método *configure(for:exam)*. Ao receber um novo exame o método *System.configure(for: exam)* é chamado, que por sua vez chama o método *Entity.configure(for: exam)* para todas as classes *Entity*. As classes do tipo *Component* podem ser instanciadas pelas classes *Entity*, em tempo de execução, de acordo com a necessidade do tipo de exame à processar. Quais componentes são necessários para processar um exame e a que entidade eles devem pertencer deve ser informação disponível na base de conhecimento. Após a instanciação dos componentes, ainda é possível realizar, caso necessário, alguma configuração destes, para que o processamento executado pelo componente esteja adequado às necessidades do tipo de exame especificado. A informação da configuração de um componente para adequação às necessidades de um tipo de exame também devem fazer parte da base de conhecimento.

Como o objetivo é a execução do processo proposto, a classe *System* é implementada de forma que possua exatamente 6 instâncias da classe *Entity*. Cada uma das 6 classes do tipo *Entity* presentes em *System* representam uma etapa do processo, sendo de grande relevância sua ordem (1 - Aquisição de Artefato, 2 –

Pré-processamento, 3 - Segmentação, 4 - Extração de atributos orientados por domínio, 5 - Classificação, 6 - Interpretação).

A execução do sistema sempre segue a mesma ordem. O método *System.run()* invoca o método *Entity.run()* de cada uma das 6 instâncias de *Entity* na ordem correta. Cada uma das instâncias de *Entity*, por sua vez chama o método *run()* de cada um de seus componentes, também em ordem pré-estabelecida. Na Figura 23 está a representação do processo proposto, implementado através do ECS.

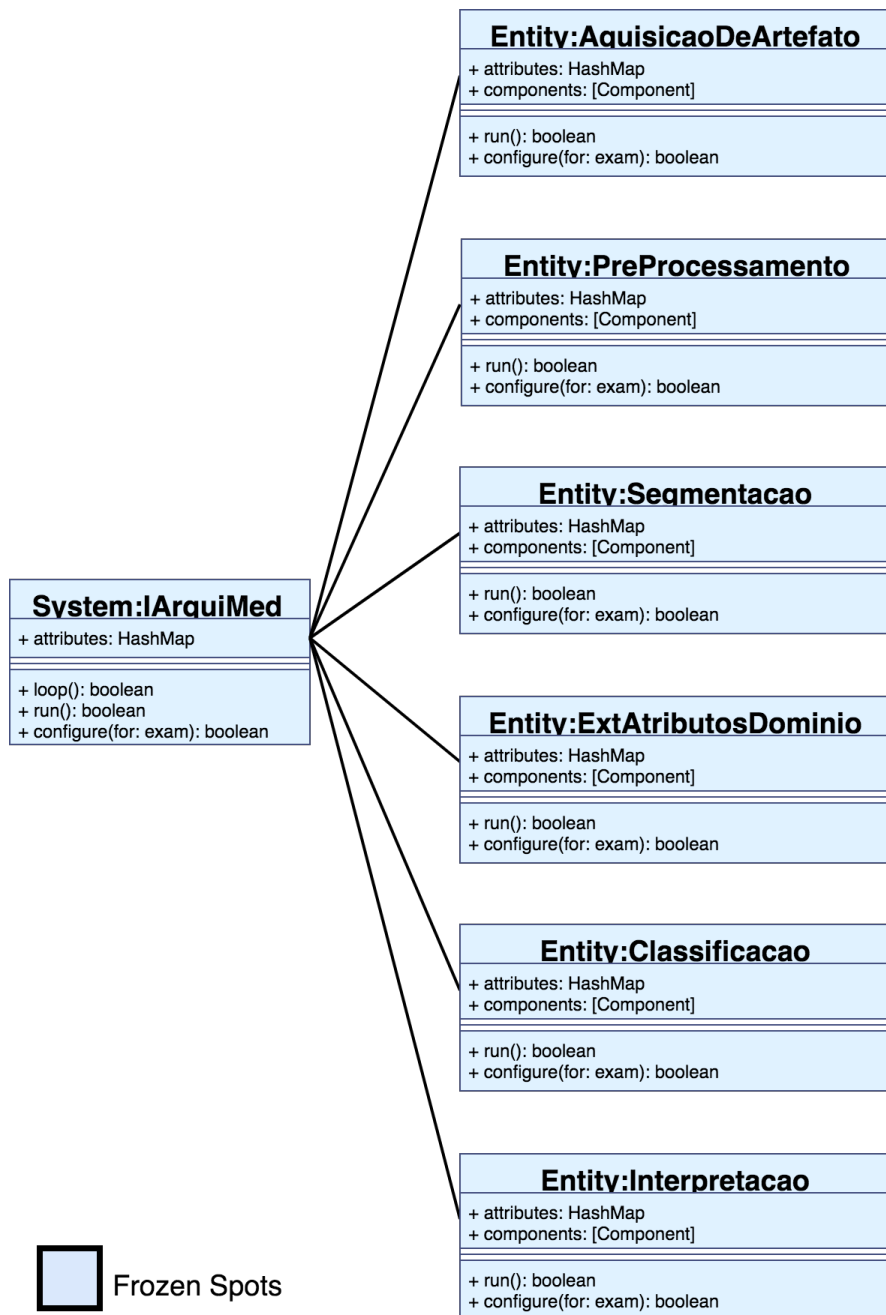


Figura 23 - Frozen-spots da implementação do processo proposto

4.3.1

Implementação Exemplo: Avaliação do Risco de Eventos Cardiovasculares Através de Ultrassonografia da Carótida com EIMC

Para apresentação deste sistema exemplo utilizando o ECS foi construído um diagrama de objetos. Derivado do diagrama de classes, o diagrama de objetos utiliza quase a mesma notação, porém o objetivo do diagrama de objetos é mostrar os objetos que foram instanciados a partir de classes. O diagrama de objetos é como uma fotografia do sistema em um certo momento de sua execução e é comumente utilizado para exemplificar a utilização das classes apresentadas no modelo de classes, expor fatos sobre elementos de modelos específicos ou para ilustrar exemplos específicos de classificadores requeridos.

No exemplo da Figura 24 representamos um sistema instanciado com a capacidade de inferir sobre imagens ultrassonográficas das artérias carótidas. Com o objetivo de simplificação do exemplo, foram escolhidos o mínimo de componentes necessários para ilustrar todas as etapas definidas no processo definido na seção 4.1, e o caso de um exame ultrassonografia da carótida com EIMC como o apresentado na Figura 19. Seguem as considerações sobre cada instância representada:

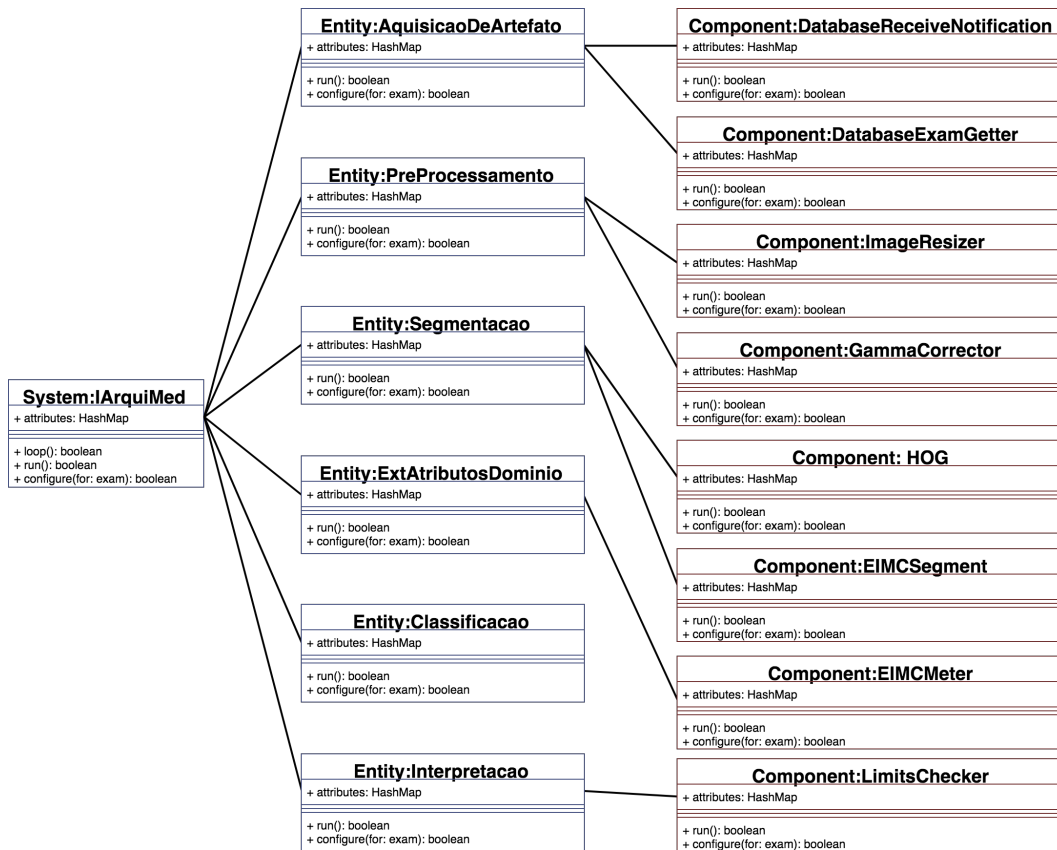


Figura 24 - Objetos instanciados do *Entity-component-system* proposto para classificação de Ultrassonografia de Carótida com EIMC

- System:iArquiMed – O objeto do tipo system implementa o processo dentro do sistema, possui a lógica que escolhe as entidades participantes do sistema e sua ordem de execução, recebe e envia mensagens das entidades e guarda as estruturas de dados que necessitem ser acessadas pelas entidades. É um objeto estático e singular no sistema (só existe uma instância do objeto do tipo System), configurado para realizar o processo desejado, encadeando a execução dos objetos do tipo Entity (entidades).
 - Entity:AquisicaoDeArtefato – Responsável por receber um novo artefato e enviá-lo, após devida adequação às estruturas de dados. Transforma objetos da base de dados em objetos definidos pelo diagrama de classes do modelo do sistema. Este componente ainda envia ao seu *owner* (seu possuidor, no caso, um objeto do tipo *System*) o novo artefato, quando devidamente transformado em um objeto do tipo *exam*.

- Component:DataBaseReceiveNotification – Especializado em receber uma notificação de uma determinada base de dados quando um novo exame é inserido e está pronto para processamento. Este componente ainda envia ao seu *owner* (seu possuidor, no caso, um objeto do tipo *System*) o novo artefato, quando devidamente transformado em um objeto do tipo *exam*.
- Component:DataBaseExamGetter –Especializado em receber um objeto de uma determinada base de dados, de tipo Exame e transformar em um objeto do modelo de classe *Exam*. Ao final deste processamento o novo objeto *Exam* é enviado ao seu *owner* (no caso, a entidade AquisicaoDeArtefato).
- Entity:PreProcessamento – Responsável por processar os dados do exame e conformá-los com características determinadas pela base de conhecimento.
 - Component:ImageResizer – Especializado em redimensionar imagens para um tamanho requerido.
 - Component:GammaCorrector – Especializado em efetuar uma correção de gama em uma imagem.
- Entity:Segmentacao – Responsável por executar os algoritmos de visão computacional
 - Component:HOG – Especializado em construir um HOG (Histogram of Oriented Gradients) (OREIFEJ, LIU, 2013) a partir de uma imagem.
 - Component: EIMCSegment – Especializado em, dado uma imagem e um HOG, extrair o segmento que representa o complexo médio-intimal (IMT) da carótida.
- Entity:ExtAtributosDominio – Responsável por executar algoritmos de extração de atributos indicados como relevantes a partir de um conhecimento prévio existente no domínio médico.
 - Component:EIMCMeter – Especializado em, dado uma imagem de ultrassom de carótida com suas estruturas anatômicas segmentadas, extrair a medida da espessura do complexo médio-intimal (IMT) na carótida primitiva.

- Entity:Classificacao – Responsável por classificar um artefato a partir de um modelo construído utilizando técnicas de aprendizado de máquina. Neste exemplo esta entidade não será implementada.
- Entity:Interpretacao – Especializado em receber todos os dados e novas informações processadas e apresentá-los ao usuário de forma adequada.
 - Component:LimitsChecker – Responsável pela recuperação das características que sejam do tipo numérico e possuam valores limite de normalidade e apresentá-las ao usuário em forma de lista com valores e quantificação de grau de normalidade.

Reforçando que as entidades do tipo *System* (iArquiMed) e *Entity* (AquisicaoDeArtefato, PreProcessamento, Segmentacao, ExtAtributosDominio, Classificacao e Interpretacao) são *frozen-spots* do sistema. Estas são customizadas através da configuração da instância porém seus aspectos são invariáveis (GAMMA et al., 1995). As classes do tipo component são *hot-spots*, devem herdar da superclasse *Component*, porém devem possuir alterações que implementem a funcionalidade de seu propósito.

Para esta tarefa foi implementado também a classe PNGImage, uma subclasse de ExamPartData, pertencente ao modelo de dados e conhecida pelos componentes processadores de imagem (GammaCorrector, HOG, EIMCsegment, EIMCMeter).

4.3.2

Implementação Exemplo: Classificador de Fundoscopias com Presença de Exsudatos Duros

Neste exemplo o sistema foi configurado para atuar como um classificador de imagens de fundoscopias possuidoras ou não de exsudatos duros, importante marcador patológico identificador de doenças como retinopatia diabética, à exemplo do sistema produzido no trabalho apresentado por Cunha e outros (2016).

A tarefa de configuração do sistema foi feita usando como ponto de partida a configuração apresentada na seção 4.3.1 deste documento, realizando à partir desta as alterações necessárias. Na Figura 25 é apresentado o diagrama de objetos do sistema configurado com esta finalidade, pronto para iniciar a classificação de uma imagem.

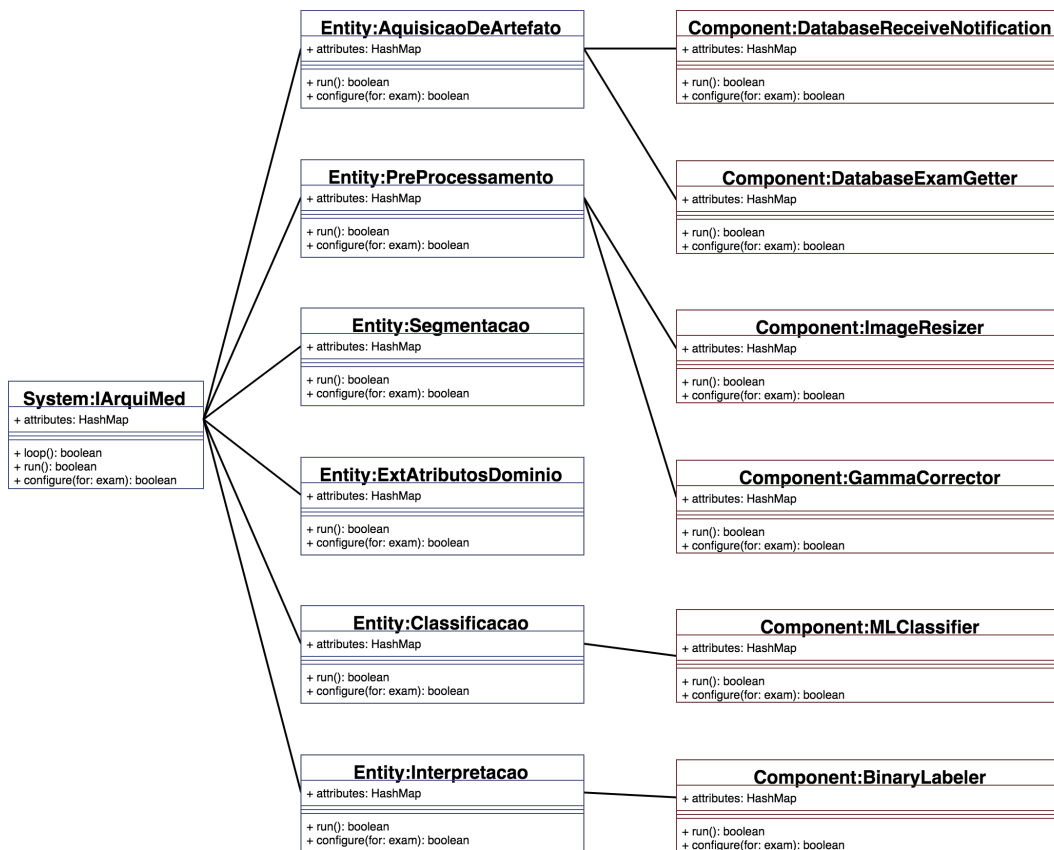


Figura 25 - Objetos instanciados do *Entity-component-system* proposto para classificação de imagens possuidoras de exsudatos duros

O pré-requisito necessário à utilização do sistema com esta configuração é a existência de um modelo classificador à ser usado, que deve estar presente na base de conhecimento. Para este fim foi usado o modelo classificador criado para o trabalho de Cunha e outros (2016).

As seguintes configurações foram realizadas:

- Entity:PreProcessamento – Configuração de seus componentes.
 - Component:ImageResizer e Component:GammaCorrector: Configurados para conformar as imagens recebidas com o novo tipo de exame, compatibilizando-as com o modelo de classificação usado.
- Entity:Segmentacao e Entity:ExtAtributosDominio
 - Todos os componentes removidos.
- Entity:Classificacao – Adicionado o componente MLClassifier e à este o modelo de classificação para exsudatos duros em fundoscopias.

- Component:MLClassifier – Especializado em, dado um artefato e um modelo de tipos compatíveis, associar o artefato à uma classe do modelo, rotulando tal artefato.
- Entity:Interpretacao – Reconfigurado.
 - Component:LimitsChecker – Removido
 - Component:BinaryLabeler – Responsável pela recuperação das características que sejam do tipo rótulo binário (possui ou não possui tal rótulo) e apresentá-las ao usuário.

É válido ressaltar que parte significativa do trabalho realizado na configuração apresentada na seção 4.3.1 foi reaproveitado para esta configuração, assim como outros artefatos necessários reaproveitado de outros trabalhos. Os componentes ImageResizer e GammaCorrector foram integralmente reaproveitados, bastando apenas configurar os parâmetros desejados para redimensionamento de imagem e correção de gamma para este caso. O componente MLClassifier foi construído à partir de um classificador binário disponível na biblioteca padrão da linguagem de programação usada para sua implementação (swift 4.1). O modelo de classificação usado foi o criado no trabalho de Cunha e outros (2016).

4.3.3

Implementação Exemplo: Classificador de Radiografias de Tórax com Presença de Pneumonias

Neste exemplo o sistema foi configurado para atuar como um classificador de imagens de radiografias de tórax possuidoras ou não dos marcadores que indicam presença de pneumonia. Em trabalho análogo ao apresentado por Macedo e outros (2012).

A tarefa de configuração do sistema foi feita usando como ponto de partida a configuração apresentada na seção 4.3.2 deste documento, realizando à partir desta as alterações necessárias. Não houve necessidade de inclusões ou exclusões de componentes para a nova tarefa, de forma que o diagrama de objetos apresentado na Figura 25 é exatamente o mesmo para a configuração com esta finalidade.

O pré-requisito necessário à utilização do sistema para esta finalidade é a existência de um modelo classificador à ser usado, que deve estar presente na base de conhecimento. Para este fim foi criado um modelo à partir de um subconjunto com 108 imagens selecionadas entre as 112.120 imagens disponibilizadas por National Institutes of Health - Clinical Center (NIH, 2017) realizando trabalho análogo ao trabalho de Macedo e outros (2012). É válido ressaltar que a qualidade do modelo classificador não é objeto desta pesquisa, sendo seus resultados não relevantes para este trabalho.

As seguintes configurações foram realizadas:

- Entity:PreProcessamento – Configuração de seus componentes.
 - Component:ImageResizer e Component:GammaCorrector: Configurados para conformar as imagens recebidas com o novo tipo de exame, compatibilizando-as com o modelo de classificação usado.
- Entity:Classificacao
 - Component:MLClassifier – Configurado para utilizar o novo modelo.

Pode-se verificar que o trabalho realizado para se configurar o sistema para realizar trabalho de mesma natureza que um trabalho anterior é consideravelmente reduzido. Neste caso específico foi necessário atualizar 1 atributo no componente ImageResizer (*destiny_size*), 1 no componente GammaCorrector (*gamma_value*), e 1 no componente MLClassifier (*model_source_location*). Todas as demais configurações são compatíveis com as utilizadas no exemplo da seção 4.3.2 deste documento.

5 Conclusão e Trabalhos Futuros

O uso de uma arquitetura baseada em *Entity-component-system*, autoconfigurável e expansível através da construção apenas de novos componentes especialistas se demonstra promissor para a construção de sistemas de apoio ao diagnóstico médico através de exames por imagem.

As abordagens tradicionais para construção de sistemas classificadores apresentam alguns desafios como o alto trabalho na realização da tarefa de geração de modelo classificador, a alta especialização destes modelos e a dificuldade de sua utilização em campanhas de produção de exames em larga escala. Estes desafios reduzem o interesse da comunidade médica à utilização de tais sistemas.

Neste contexto, este trabalho propôs uma solução para apoio ao diagnóstico médico de exames por imagem capaz de:

- a) Suportar a classificação e priorização de exames em batch (lote, batelada).
 - Através dos componentes observadores, a cada exame ainda não processado (classificado como novo exame), o sistema inicia um novo processamento, atualizando a base de dados de exames, inclusive com as novas informações processadas, ao final do processamento.
- b) Suportar a classificação através de modelos gerados por aprendizagem de máquina e acumular diversos modelos de classificação, utilizando-os de acordo com o contexto.
 - Utilizando uma base de conhecimento detentora de diversos modelos classificadores, estes podem ser selecionados para utilização de acordo com o contexto, possibilitando um mesmo sistema atender à vários contextos diferentes.
- c) Suportar algoritmos de visão computacional para segmentação da imagem de exame em suas subestruturas anatômicas, aumentando a

acurácia do classificador e reaproveitando em diferentes contextos os algoritmos de avaliação de um mesmo tipo de subestrutura.

- O mesmo algoritmo de segmentação pode ser usado em diferentes contextos compatíveis, viabilizando e facilitando o reuso das implementações realizadas .
 - A acurácia dos algoritmos classificadores é proporcional à quantidade de dados dos exemplos utilizados (DOUGLAS et al., 2011) , como a segmentação diminui a quantidade de componentes independentes em amostras, a utilização de componentes segmentados tendem a aumentar a acurácia.
 - A segmentação da imagem de exame permite também a extração de atributos orientados por domínio, muitas vezes impossíveis de serem adquiridas sem essa técnica, como comprimento, área, volume e densidade de algumas subestruturas anatômicas.
- d) Suportar algoritmos de detecção de anomalias, reduzindo o conjunto de dados nos *datasets* geradores de conhecimento às imagens normais e gerando não só sistemas de reconhecimento de anomalias como métricas de qualificação de anomalias.
- Algoritmos de detecção de anomalias reduz o trabalho na coleta de exemplos para geração do modelo classificador, dado que apenas exemplos ‘normais’, ou na linguagem do contexto ‘saudáveis’ são necessários para geração deste tipo de modelo.
 - Com esta técnica é possível também gerar uma métrica de ‘distanciamento da normalidade’, bastante interessante para o contexto sugerido, essa métrica pode apoiar a priorização ou a urgência no atendimento de pacientes.
- e) Suportar a classificação através de inserção de algoritmos representantes de conhecimento de domínio, substituindo a utilização de modelos por tais algoritmos.
- Conforme relatado no item (c) desta seção, esta abordagem permite o aproveitamento de conhecimento do domínio médico para geração de classificadores que não dependam da construção de modelos de classificação, o que, em alguns casos, pode reduzir

- o trabalho na geração de processadores de exames para a classificação dos mesmos.
- f) Poder evoluir através de adição de componentes sem a necessidade de se modificar os módulos já construídos e validados.
- O ECS se apresentou como um paradigma de programação que promove o desacoplamento e aumenta a manutenibilidade dos sistemas que o aplicam.
 - O domínio escolhido sugere a necessidade de utilização de técnicas diversas para aquisição da informação em seus diversos subdomínios, número tão grande quanto a quantidade de doenças identificáveis através de exames médicos por imagem.
 - Mostra-se desejável a maximização de reaproveitamento de conhecimento para diferentes contextos, assim como a possibilidade de se acumular esse conhecimento gerado através dos artefatos produzidos (algoritmos, modelos classificadores, parametrizações, etc.), em um único e acessível local.
- g) Aceitar diferentes geradores de conhecimento, de diversos tipos e para diferentes finalidades, convidando a comunidade científica a agrupar o trabalho realizado para o diagnóstico automatizado em sistemas que executem a tarefa para diversos subdomínios, utilizando diferentes abordagens.

Neste trabalho foi desenvolvido uma arquitetura para apoiar a construção de um sistema que possua todas as características desejadas.

5.1

Contribuições

A seguir são apresentadas as contribuições diretas resultantes do desenvolvimento deste trabalho:

1. **Uma abordagem para a construção de um sistema de apoio e priorização ao diagnóstico médico por imagem em batch:** Foi proposta uma abordagem para apoiar a construção de um sistema que suprisse essas necessidades da comunidade médica.

2. **A construção de uma arquitetura de software que suportasse a implementação proposta:** Foi construída tal arquitetura de software, baseado na arquitetura *Entity-component-system*, implementando todas as características descritas nos itens de (a) até (f) da seção 5 deste documento.
3. **O desenvolvimento parcial do sistema proposto:** Para fins de avaliação, o sistema proposto foi parcialmente implementado utilizando a arquitetura construída. Como o estudo dos algoritmos que executam as tarefas sugeridas na arquitetura não fazem parte deste trabalho, estes não foram inteiramente implementados. Estes deverão ser implementados para o perfeito funcionamento do sistema proposto e está listado em trabalhos futuros.

5.2

Trabalhos Futuros

As contribuições apresentadas anteriormente são um esforço para prover aos desenvolvedores de sistemas de apoio ao diagnóstico médico maior aparato ferramental para construção de sistemas desta natureza. É também uma tentativa de se gerar um agrupamento dos esforços da comunidade desenvolvedora em matéria que pode vir a ser a solução para problemas de saúde pública como: identificação precoce de doenças, atenção a população de baixa renda e a áreas com baixa densidade de médicos por habitante em qualquer especialidade.

No que se refere ao sistema gerado, diversos pontos podem ser melhorados e a seguir são apresentados os trabalhos futuros que podem ser realizados como desdobramentos do trabalho proposto nesta dissertação.

1. **Implementação da autoconfiguração:** Evoluir a implementação do sistema para suportar autoconfiguração por contexto, selecionando automaticamente os componentes necessários para a avaliação de cada exame recebido.
2. **Implementação de interfaces amigáveis:** Implementação de interfaces gráficas amigáveis para utilização do sistema de apoio diagnóstico.

3. **Avaliação com novos estudos de caso:** Efetuar avaliação comparando com sistemas operando no mesmo ambiente de execução. Para este estudo deve-se utilizar um sistema classificador padrão para realizar a tarefa em dois contextos e compará-los com a execução das mesmas no sistema proposto, configurando-o para tal. Os contextos sugeridos são: Mamografia (câncer de mama), Radiografia Torácica (Pneumonia na Infância) e Fundoscopia Ocular (Retinopatia Diabética) por já existirem trabalhos realizados com estas bases (MACEDO et al., 2012; GARDEZI; ELTOUKHY; FAYE, 2017; CUNHA et al., 2016) e estas estarem disponíveis para uso acadêmico.
4. **Implementação de um repositório de conhecimento:** Implementar um repositório de conhecimento público para a comunidade desenvolvedora, médica e científica, para que o conhecimento persistido em forma de modelos de classificação, matrizes de covariância, algoritmos de segmentação e quantificação de atributos, entre outros, possam ser centralizados e compartilhados com a comunidade.

6

Referências bibliográficas

ABRAMOFF, M; NIEMEIJER, M. et al. **Evaluation of a system for automatic detection of diabetic retinopathy from color fundus photographs in a large population of patients with diabetes.** In: Diabetes care, 2008. (document), volume 31(2), p. 193-198.

ACHER, Mathieu et al. **Extraction and evolution of architectural variability models in plugin-based systems.** Software & Systems Modeling, v. 13, n. 4, p. 1367-1394, 2014.

AGGARWAL, Charu C. An introduction to outlier analysis. In: **Outlier analysis.** Springer, New York, NY, 2013. p. 1-40.

AHA, David W. Editorial. In: **Lazy learning.** Springer, Dordrecht, 1997. p. 7-10.

AHA, David W.; KIBLER, Dennis; ALBERT, Marc K. **Instance-based learning algorithms.** Machine learning, v. 6, n. 1, p. 37-66, 1991.

ALATALO, Toni. **An entity-component model for extensible virtual worlds.** IEEE Internet Computing, v. 15, n. 5, p. 30-37, 2011.

ALMEIDA, Manoel Roberto Aguirre. **Sistema híbrido neuro-fuzzy-genético para mineração automática de dados.** Rio de Janeiro: PUC. Dept. de Engenharia Elétrica, 2004.

AMERI EKHTIARABADI, Afshin et al. **A general framework for incremental processing of multimodal inputs.** In: Proceedings of the 13th international conference on multimodal interfaces. ACM, 2011. p. 225-228.

ANS, **Mapa Assistencial da Saúde Complementar**, Disponível em: <<http://www.ans.gov.br/>>. Acesso em: 20 ago. 2017.

APPLE Inc. (2017). **About Swift.** Disponível em: <<https://swift.org/about/#swiftorg-and-open-source>>. Acessado em 02.12.2017.

APPLE Inc., **Build more intelligent apps with machine learning**, Disponível em: <[https:// developer.apple.com/machine-learning/](https://developer.apple.com/machine-learning/)>. Acesso em: 02 jan. 2018.

ARAÚJO, R. T. S. **Detecção de Manchas de Óleo na Superfície do Mar em Imagens de Radar de Abertura Sintética. Dissertação de Mestrado – Universidade Federal do Ceará, 2004.**

AYED, Ismail Ben et al. **Graph cuts with invariant object-interaction priors: application to intervertebral disc segmentation.** In: Biennial International Conference on Information Processing in Medical Imaging. Springer, Berlin, Heidelberg, 2011. p. 221-232.

BAATZ, M.; SCHAPE, A. **Multiresolution Segmentation: An Optimization Approach for High Quality Multi-Scale Image Segmentation.** In: Strobl, J., Blaschke, T. and Griesbner, G., Eds., *Angewandte Geographische Informations-Verarbeitung, XII*, Wichmann Verlag, Karlsruhe, Germany, 2000. p.12-23.

BALLARD, D; BROWN, C. **Computer Vision.** NY: Prentice Hill, 1982.

BLEAU, André; LEON, L. Joshua. **Watershed-based segmentation and region merging.** *Computer Vision and Image Understanding*, v. 77, n. 3, p. 317-370, 2000.

BOTTOU, L. **Large-scale machine learning with stochastic gradient descent.** In: *Proceedings of COMPSTAT'2010*. Physica-Verlag HD, 2010. p. 177-186.

BRAGA, A. de P.; CARVALHO, ACPLF; LUDERMIR, Teresa Bernarda. **Redes neurais artificiais: teoria e aplicações.** Livros Técnicos e Científicos, 2000.

BUSCHMANN et al. **Pattern-Oriented Software Architecture - Volume 1: A System of Patterns.** Wiley Publishing, 1996.

CHEESEMAN, P. et al. **Autoclass III.** Program available from NASA Ames Research Center: Research Institute for Advanced Computer Science, 1990.

CHERUBINI, Andrea et al. **A unified multimodal control framework for human–robot interaction.** *Robotics and Autonomous Systems*, v. 70, p. 106-115, 2015.

CORREA, Z.; EAGLE, R. **Aspectos patológicos da retinopatia diabética.”** *Revista de Oftalmologia de São Paulo*, v.68 (3), p. 410-414, 2005.

CUNHA, F.; VENIERIS, R. **Aprendizagem de máquina na tarefa de classificação de doenças a partir de imagens de fundo de olho.** 2015.

CUNHA, F.; VENIERIS, R.; VIANA M.; LUCENA, C. J. P. **Aprendizado de máquina na classificação de retinopatia diabética em fundoscopia**. 2016.

DA SILVA MOREIRA, João Vitor; LIMA, Mário Oliveira; MARCIANO, Fernanda Roberta. ENGENHARIA BIOMÉDICA APLICADA À REABILITACAO EM PÓS-OPERATÓRIO DE MASTECTOMIA. **Revista Univap**, v. 22, n. 40, p. 903, 2017.

DataSUS. SIGTAP - **Sistema de Gerenciamento da Tabela de Procedimentos, Medicamentos e OPM do SUS**. [acessado 30-08-2017]. Disponível em: <<http://sigtap.datasus.gov.br/tabela-unificada/app/sec/inicio.jsp>>

DE NUNES, Ingrid Oliveira; CARVALHO, Dárlinton Barbosa Feres; DE LUCENA, Carlos José Pereira. **Estudo sobre Algoritmos de Classificação para o Referenciamento de Gestantes de Alto-risco**, 2009.

DIAS, T. M. R. et al. **IDENTIFICATION OF DISEASE PATTERNS IN DATA MINING TECHNIQUES**. Blucher Mechanical Engineering Proceedings, v. 1, n. 1, p. 4634-4643, 2012.

DIAS, T. M. R. et al. **IDENTIFICATION OF DISEASE PATTERNS IN DATA MINING TECHNIQUES**. **Blucher Mechanical Engineering Proceedings**, v. 1, n. 1, p. 4634-4643, 2012.

DILL, Vanderson. **Segmentação de hipocampo em imagens de ressonância magnética utilizando seleção de atlas por meta-informações**. 2013. Dissertação de Mestrado. Pontifícia Universidade Católica do Rio Grande do Sul.

DOUGLAS, Pamela K. et al. **Performance comparison of machine learning algorithms and number of independent components used in fMRI decoding of belief vs. disbelief**. **Neuroimage**, v. 56, n. 2, p. 544-553, 2011.

EDWARDS, Gareth J.; COOTES, Timothy F.; TAYLOR, Christopher J. **Face recognition using active appearance models**. In: European conference on computer vision. Springer, Berlin, Heidelberg, 1998. p. 581-595.

EL-BAZ, Ayman; JIANG, Xiaoyi; SURI, Jasjit S. (Ed.). **Biomedical Image Segmentation: Advances and Trends**. CRC Press, 2016.

FAYYAD, Usama; PIATETSKY-SHAPIRO, Gregory; SMYTH, Padhraic. **From data mining to knowledge discovery in databases**. **AI magazine**, v. 17, n. 3, p. 37, 1996.

FÉLIX, John Hebert da Silva. **Sistema de visão computacional para detecção e quantificação de enfisema pulmonar**. 2007, Tese de Doutorado.

FERREIRA, Sueli Mara Soares Pinto. **Novos paradigmas da informação e novas percepções do usuário**. Ciência da Informação, v. 25, n. 2, 1996.

FILHO, M., Ogê; NETO, Hugo Vieira. **Processamento digital de imagens**, Brasport, 1999.

FIOCRUZ. **Campanhas de exames regionais e nacionais**. [acessado 10-09-2017]. Disponível em: <<https://portal.fiocruz.br/pt-br/search/site/campanha-exames>>

FISCHBACH, Martin; WIEBUSCH, Dennis; LATOSCHIK, Marc Erich. **Semantic Entity-Component State Management Techniques to Enhance Software Quality for Multimodal VR-Systems**. IEEE Transactions on Visualization and Computer Graphics, v. 23, n. 4, p. 1342-1351, 2017.

FISCHBACH, Martin. **Software techniques for multimodal input processing in realtime interactive systems**. In: Proceedings of the 2015 ACM on International Conference on Multimodal Interaction. ACM, 2015. p. 623-627.

FOWLER, Martin. **Patterns of enterprise application architecture**. Addison-Wesley Longman Publishing Co., Inc., 2002.

GAMMA, Erich et al. **Design patterns: Abstraction and reuse of object-oriented design**. In: European Conference on Object-Oriented Programming. Springer, Berlin, Heidelberg, 1993. p. 406-431.

GAMMA, Erich. **Design patterns: elements of reusable object-oriented software**. Pearson Education India, 1995.

GARDEZI, Syed Jamal Safdar; ELTOUKHY, Mohamed Meselhy; FAYE, Ibrahima. **Machine Learning Applications in Breast Cancer Diagnosis**. In: Handbook of Research on Machine Learning Innovations and Trends. IGI Global, 2017. p. 465-490.

GEGO, Governo do Estado de Goiás. **De Olho na Visão**. 2016. Disponível em <<http://www.casacivil.go.gov.br/post/busca?campoBusca=de+olho+na+visao>>. Acesso em: 05 abr. 2017.

GESP, Governo Estadual De São Paulo. **Carretas de mamografia oferecem exames gratuitos na capital e interior**. 2016. Disponível em: <<http://www.saopaulo.sp.gov.br/spnoticias/ultimas-noticias/carretas-de-mamografia-oferecem-exames-gratuitos-na-capital-e-interior/>> Acesso em: 05 abr. 2017.

GOLDBERG, David E.; HOLLAND, John H. **Genetic algorithms and machine learning**. Machine learning, v. 3, n. 2, p. 95-99, 1988.

GONZALEZ, R. C.; WOODS, R. **Digital Image Processing**". 3a. ed. EUA: Prentice Hall, 2002.

GREENSPAN, Hayit; VAN GINNEKEN, Bram; SUMMERS, Ronald M. **Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique**. IEEE Transactions on Medical Imaging, v. 35, n. 5, p. 1153-1159, 2016.

GUEDES, Gilleanes TA. **UML: uma abordagem prática**. Novatec Editora, 2008.

GUPTA, Manish et al. Outlier detection for temporal data: A survey. **IEEE Transactions on Knowledge and Data Engineering**, v. 26, n. 9, p. 2250-2267, 2014.

HAMZA, Md Ameer. **Meerkat: Extending entity-based programming to networked games**. 2012. Tese de Doutorado. Queen's University (Canada).

HARALICK, R. M. **Computer and Robot Vision**. New York: Addison-Wesley, 1972.

HASTIE, Trevor; TIBSHIRANI, Robert. Classification by pairwise coupling. In: **Advances in neural information processing systems**. 1998. p. 507-513.

HAYKIN, Simon S. et al. (Ed.). **Kalman filtering and neural networks**. New York: Wiley, 2001.

HODGE, Victoria; AUSTIN, Jim. A survey of outlier detection methodologies. **Artificial intelligence review**, v. 22, n. 2, p. 85-126, 2004.

HOFFMAN, Donald D.; RICHARDS, Whitman A. **Parts of recognition – Cognition**", v. 18, n. 1, p. 65-96, 1984.

HSU, C.; CHANG, C.; LIN, C. **A practical guide to support vector classification**." Technical Report, NTU, 2003.

JAN, J. **Medical Image Processing Reconstruction and Restoration: Concepts and Methods**". Danvers, MA, U.S.A: CRC Press, 2006.

JOHN, George H.; LANGLEY, Pat. Estimating continuous distributions in Bayesian classifiers. In: **Proceedings of the Eleventh conference on Uncertainty in artificial intelligence**. Morgan Kaufmann Publishers Inc., 1995. p. 338-345.

JOHNSON, Ralph E.; FOOTE, Brian. Designing reusable classes. **Journal of object-oriented programming**, v. 1, n. 2, p. 22-35, 1988.

JOHNSON, Ralph E.; RUSSO, Vincent. **Reusing object-oriented designs**. Department of Computer Science, University of Illinois at Urbana-Champaign, 1991.

JÚNIOR, Francisco das Chagas Monteiro et al. **Prevalência de aterosclerose subclínica e reclassificação de risco cardiovascular pela medida da espessura íntimo-medial carotídea em pacientes hipertensos ambulatoriais**. Revista Portuguesa de Cardiologia, v. 32, n. 12, p. 975-980, 2013.

KEERTHI, S.. Sathiya et al. Improvements to Platt's SMO algorithm for SVM classifier design. **Neural computation**, v. 13, n. 3, p. 637-649, 2001.

LANGE, Patrick et al. **Virtual reality for simulating autonomous deep-space navigation and mining**. In: 24th International Conference on Artificial Reality and Telexistence (ICAT-EGVE 2014). 2014.

LATOSCHIK, M.; FISCHBACH, M. **Engineering Variance: Software Techniques for Scalable, Customizable, and Reusable Multimodal Processing**. In Proceedings of the HCI International Conference, HCII, pages 308–319. Springer, 2014.

LATOSCHIK, Marc Erich; TRAMBEREND, Henrik. **Simulator X: A scalable and concurrent architecture for intelligent realtime interactive systems**. In: Virtual Reality Conference (VR), 2011 IEEE. IEEE, 2011. p. 171-174.

LATOSCHIK, Marc Erich. **A user interface framework for multimodal VR interactions**. In: Proceedings of the 7th international conference on Multimodal interfaces. ACM, 2005. p. 76-83.

LECUN, Y.; KAVUKCUOGLU, K.; FARABET, Clément. **Convolutional networks and applications in vision**. In: Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on. IEEE, 2010. p. 253-256.

LI, Juhua et al. **Promotion of Educational Effectiveness by Translation-based Programming Language Learning Using Java and Swift**. 2017.

LIN, Jimmy; DEMNER-FUSHMAN, Dina. **The role of knowledge in conceptual retrieval: a study in the domain of clinical medicine**. In: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2006. p. 99-106.

LLOPIS, Noel. **PROGRAMMING-THE INNER PRODUCT-Data-Oriented Design**. Game Developer, v. 16, n. 8, p. 43, 2009.

MACEDO, Sanderson Oliveira de et al. **Desenvolvimento de um sistema de auxílio ao diagnóstico de pneumonia na infância utilizando visão computacional**”, 2012.

MATTSSON, Michael. Object-oriented frameworks. **Licentiate thesis**, 1996.

MENDES, Charles Corrêa. **Classificação de imagens de retina para auxílio a diagnóstico de retinopatia diabética via detecção de exsudatos duros, com base em classificadores robustos, máquinas de vetores de suporte e algoritmos com treinamento de dicionários**. 2017.

MINA, Andrea et al. **Mapping evolutionary trajectories: Applications to the growth and transformation of medical knowledge**. Research policy, v. 36, n. 5, p. 789-806, 2007.

MIRZAEI, Golrokh; ADELI, Anahita; ADELI, Hojjat. **Imaging and machine learning techniques for diagnosis of Alzheimer’s disease**. Reviews in the Neurosciences, v. 27, n. 8, p. 857-870, 2016.

MITCHELL, Melanie. **An introduction to genetic algorithms**. MIT press, 1998.

MOSKOVITCH, Robert; SHAHAR, Yuval. Medical temporal-knowledge discovery via temporal abstraction. In: **AMIA annual symposium proceedings**. American Medical Informatics Association, 2009. p. 452.

NIH, National Institutes of Health - Clinical Center, **NIH Chest X-ray Dataset of 14 Common Thorax Disease Categories**, 2017. Disponível em: <<https://www.nih.gov/news-events/news-releases/nih-clinical-center-provides-one-largest-publicly-available-chest-x-ray-datasets-scientific-community>>, Acessado em: 01 fev. 2018.

NOVER, Arno. **O fundo do olho: métodos de exames e achados típicos**. Manole, 2000.

OREIFEJ, Omar; LIU, Zicheng. Hon4d: **Histogram of oriented 4d normals for activity recognition from depth sequences**. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2013. p. 716-723.

OTHMAN, Nadia; DORIZZI, Bernadette; GARCIA-SALICETTI, Sonia. OSIRIS: An open source iris recognition software. **Pattern Recognition Letters**, v. 82, p. 124-131, 2016.

PARDINI, Hermes. **Manual de Exames de Imagem**, 2014.

PIATETSKY-SHAPIRO, Gregory. **Knowledge discovery in real databases**: A report on the IJCAI-89 Workshop. *AI magazine*, v. 11, n. 4, p. 68, 1990.

PLATT, John. **Sequential minimal optimization: A fast algorithm for training support vector machines**. 1998.

PRATT, W. K. **Digital Image Processing**. 3 ed. New York: John Wiley and Sons, Inc, 2001.

PREE, Wolfgang. Hot-spot-driven framework development. **Framework**, v. 2, p. B1, 1995.

PROGRESSO, O. **Campanha de prevenção do glaucoma atende mais de 3 mil na Capital**. 2015. Disponível em <<http://www.progresso.com.br/dia-a-dia/campanha-de-prevencao-do-glaucoma-atende-mais-de-3-mil-na-capital>>. Acesso em: 06 abr. 2017.

RASMUSSEN, C. E.; WILLIAMS, C. KI. **Gaussian processes for machine learning**. Cambridge: MIT press, 2006.

ROELKE, Leonard Hermann et al. **Correlação da espessura médio-intimal das carótidas primitivas proximal e distal**. *Arq Bras Cardiol*, v. 101, n. 3, p. 211-6, 2013.

ROTH, Louis M.; WILLIS, Edwin R. **Parthenogenesis in cockroaches**. *Annals of the Entomological Society of America*, v. 49, n. 3, p. 195-204, 1956.

RSNA Org., **Artists Sought for RSNA 2017 Image**, Disponível em: <<http://www.rsna.org/news.aspx?id=22764>>. Acesso em: 01 set. 2017.

RUCK, Dennis W. et al. **The multilayer perceptron as an approximation to a Bayes optimal discriminant function**. *IEEE Transactions on Neural Networks*, v. 1, n. 4, p. 296-298, 1990.

RUIZ, Enrique Vidal. **An algorithm for finding nearest neighbours in (approximately) constant average time**. *Pattern Recognition Letters*, v. 4, n. 3, p. 145-157, 1986.

RUSSO, Fabrizio; RAMPONI, Giovanni. **A fuzzy filter for images corrupted by impulse noise**. IEEE Signal Processing Letters, v. 3, n. 6, p. 168-170, 1996.

SBC. **Grandes Desafios da Computação no Brasil**. Disponível em: <<http://www.sbc.org.br/documentos-da-sbc/send/141-grandes-desafios/802-grandesdesafiosdacomputaonobrasil>> Acessado em: 14-02-2018

SBC. **II Seminário sobre Grandes Desafios da Computação no Brasil, Integração com a Indústria e uma Perspectiva para 2020**. Disponível em: <<http://www.sbc.org.br/documentos-da-sbc/send/141-grandes-desafios/801-grandesdesafios-pt-v3-2012>> Acessado em: 14-02-2018

SBIS, **Lista de Sistemas Certificados SBIS-CFM**. Disponível em: <<http://www.sbis.org.br/lista-de-sistemas-certificados>>. Acesso em: 10 mar. 2017.

SCHEFFER, Mário. **Demografia médica no Brasil (Vol. 2)**. São Paulo: Conselho Federal de Medicina e Conselho Regional de Medicina do Estado de São Paulo, 2013.

SHOJAEI, R.; ALIREZA, J.; BABYN, P. **Automatic lung segmentation in ct images using watershed transform**. IEEE International Conference on Image Processing ICIP, v. 2, n. 1270-1273, 2005.

SILVA, Aristófanos C.; CARVALHO, Paulo Cezar P. Sistema de análise de nódulo pulmonar. In: **II Workshop de Informática aplicada a Saúde, Itajai, Universidade de Itajai**. 2002.

SIMÕES, João Carlos. **No Brasil faltam médicos: mito ou realidade?** Revista do Médico Residente, v. 13, n. 4, 2011.

SIMON, Dan. **Evolutionary optimization algorithms**. John Wiley & Sons, 2013.

TOENNIES, Klaus D.; KÖNIG, Tim; GLOGER, Oliver. **Domain knowledge for level set segmentation in medical imaging**. Biomed Image Segmentation Adv Trends, p. 51-74, 2016.

UML: **Unified Modeling Language Specification**, Version 2.0, OMG, Disponível em: <<http://www.omg.org/uml/>>. Acessado em: 21-02-2015.

WANG, Dayong et al. **Deep learning for identifying metastatic breast cancer**. arXiv preprint arXiv:1606.05718, 2016.

WATSON, Ian; MARIR, Farhi. **Case-based reasoning: A review**. The knowledge engineering review, v. 9, n. 4, p. 327-354, 1994.

WEBB, W. Richard; HIGGINS, Charles B. **Thoracic imaging: pulmonary and cardiovascular radiology**. Lippincott Williams & Wilkins, 2010.

WIEBUSCH, Dennis; LATOSCHIK, Marc Erich. **Decoupling the entity-component-system pattern using semantic traits for reusable realtime interactive systems**. In: Software Engineering and Architectures for Realtime Interactive Systems (SEARIS), 2015 IEEE 8th Workshop on. IEEE, 2015. p. 25-32.

WITTEN, Ian H. et al. **Data Mining: Practical machine learning tools and techniques**. Morgan Kaufmann, 2016.

WOLFF, Klaus; JOHNSON, Richard Allen. **Fitzpatrick's color atlas and synopsis of clinical dermatology**. McGraw Hill, 2009.

ZAKS, Maxim., **When not to use ECS**, Disponível em: <<https://hackernoon.com/when-not-to-use-ecs-48362c71cf47>>. Acesso em: 01 fev. 2017.

ZHOU, Shang-Ming et al. **Defining disease phenotypes in primary care electronic health records by a machine learning approach: a case study in identifying rheumatoid arthritis**. PloS one, v. 11, n. 5, p. e0154515, 2016.

BARRETO, C. G.; FUKS, H.; LUCENA, C. J. P. **Agregando Frameworks em uma Arquitetura Baseada em Componentes no Ambiente AulaNet**. In: Anais do 5º Workshop de Desenvolvimento Baseado em Componentes-WDBC. 2005. p. 7-9.

BASIL, V. R. **Software modeling and measurement: the Goal/Question/Metric paradigm**. 1992.

BASS, Len. Paul Clements, and Rick Kazman. **Software architecture in practice**, 2003.

BENBASSAT, Fernando; BORBA, Paulo; TEIXEIRA, Leopoldo. **Safe Evolution of Software Product Lines: Feature Extraction Scenarios**. In: Software Components, Architectures and Reuse (SBCARS), 2016 X Brazilian Symposium on. IEEE, 2016. p. 11-20.

CAROLI, P. **Uma Metodologia de Projeto de Software Orientado a Objetos**. PUC-RIO. Rio de Janeiro, 1999.

CERQUEIRA, S. L. R. et al. **Plataforma GeoRisc Engenharia da Computação Aplicada à Análise de Riscos Geo-ambientais**. PUC-RIO. Rio de Janeiro, 2009.

HUBER, Nikolaus et al. **Modeling run-time adaptation at the system architecture level in dynamic service-oriented environments**. Service Oriented Computing and Applications, v. 8, n. 1, p. 73-89, 2014.

IBM. An architectural blueprint for autonomic computing – Technical Report. 2003.

JACOBSON, Ivar et al. **The unified software development process**. Reading: Addison-Wesley, 1999.

KAWAMURA, Takao. **Interpretação de um teste sob a visão epidemiológica: eficiência de um teste**. Arquivos Brasileiros de Cardiologia, v. 79, n. 4, p. 437-441, 2002.

KRAFZIG, Dirk et al. **Service-Oriented Architecture Best Practices (The Coad Series)**. 2004.

LIMA, Gleydson de Azevedo Ferreira. **Uma abordagem para evolução e reconciliação de linhas de produtos de software clonadas**. 2014.

LUCENA, C. J. P. D. **Inteligência Artificial e Engenharia de Software**. Rio de Janeiro: Zahar, 1987.

MEDEIROS, Flávio et al. **A comparison of 10 sampling algorithms for configurable systems**. In: Proceedings of the 38th International Conference on Software Engineering. ACM, 2016. p. 643-654.

MONARD, Maria Carolina; BARANAUSKAS, José Augusto. Conceitos sobre aprendizado de máquina. **Sistemas inteligentes-Fundamentos e aplicações**, v. 1, n. 1, p. 32, 2003.

PAUL, I. **Firefox will stop supporting plugins by end of 2016, following Chrome's lead**. *PCWorld*. IDG. Retrieved 20 October 2016.

PETRISOR, Teodora-Roxana. **An empirical evaluation of using the Swift language as the underlying technology of RESTful APIs**. Undergraduate Honors Thesis. TH Köln–University of Applied Sciences, Campus Gummersbach, 2016.

PICCIONI, Marco; FURIA, Carlo A.; MEYER, Bertrand. **An empirical study of API usability**. In: Empirical Software Engineering and Measurement, 2013 ACM/IEEE International Symposium on. IEEE, 2013. p. 5-14.

POHL, Klaus; BÖCKLE, Günter; VAN DER LINDEN, Frank J. **Software product line engineering: foundations, principles and techniques**. Springer Science & Business Media, 2005.

QUINLAN, J. Ross. **Generating production rules from decision trees.** In: ijcai. 1987. p. 304-307.

REBOUÇAS, Marcel et al. **An empirical study on the usage of the swift programming language.** In: Software Analysis, Evolution, and Reengineering (SANER), 2016 IEEE 23rd International Conference on. IEEE, 2016. p. 634-638.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence - A Modern Approach.** New Jersey, USA: Prentice Hall, 2009. 1152 p.

STAKE, Robert E. **Case Studies** In: Denzin N. and Lincoln, Y.(Ed.). Handbook of qualitative research (pp. 435-455). 2000.

UNITY TECHNOLOGIES. **Unity.** <<http://www.unity3d.com>>, 2015. Last accessed 2016-12-01.

7 Apêndices

Este capítulo apresenta informações adicionais relevantes para o melhor entendimento das motivações e decisões tomadas pelo autor para o desenvolvimento deste trabalho.

Na seção 7.1 – Apêndice A, será apresentada a situação da cobertura médica oftalmológica no Brasil no ano de 2014.

Na seção 7.2 – Apêndice B, serão apresentadas algumas considerações complementares sobre a abordagem para resolução do problema tomada pelo autor, o ato médico e seus fundamentos legais.

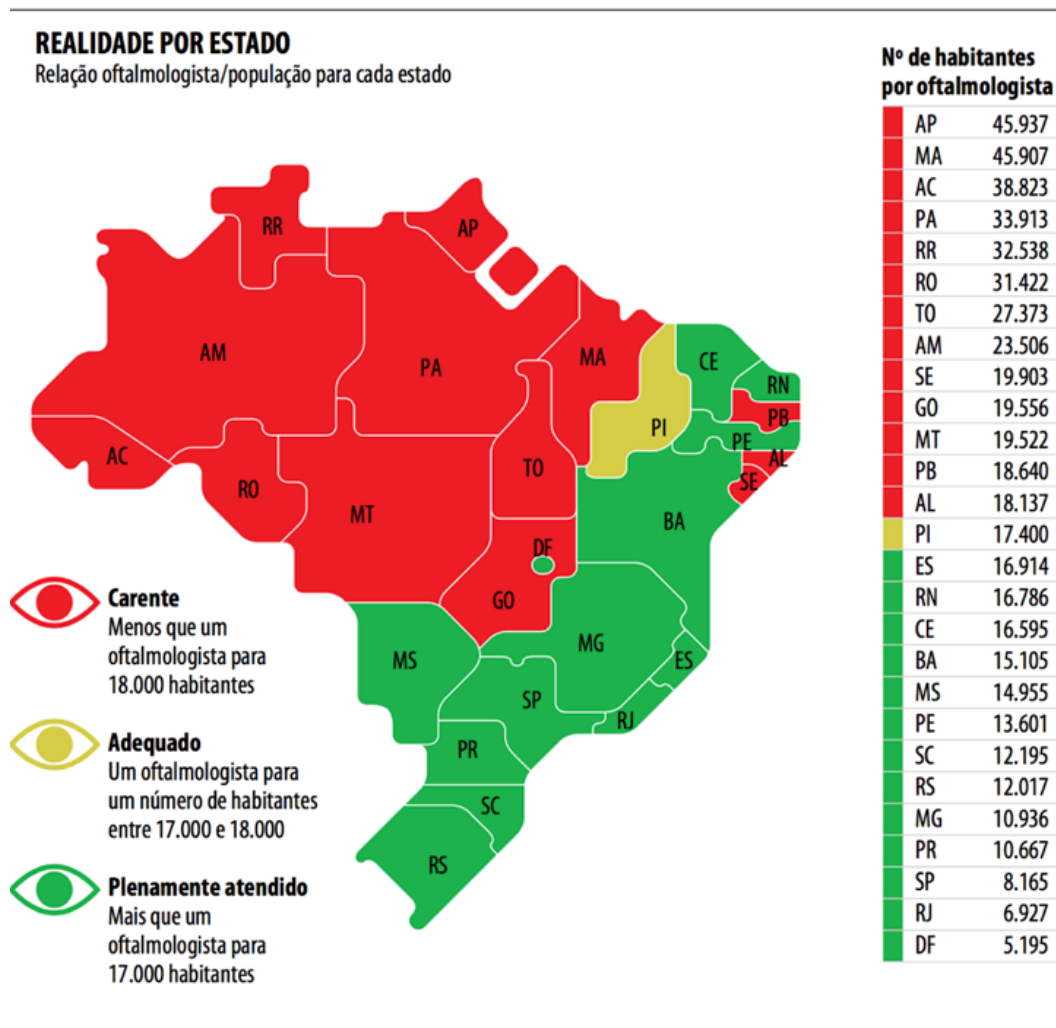
7.1

Apêndice A – Situação da Cobertura Médica Oftalmológica no Brasil

A análise de exames médicos e seu diagnóstico exato é a mais eficaz atividade médica para o tratamento precoce de doenças.

O Brasil não possui uma distribuição ideal de médicos por seu território. Enquanto as grandes capitais do país estão próximas da densidade mínima de médicos de 2,0‰ (dois médicos por mil habitantes), indicada pelo Conselho Federal de Medicina (CFM), algumas regiões possuem taxas de até 0,6‰ médicos (SIMÕES, 2011). Quando se aplica à essas taxas o percentual de médicos por especialidade é possível verificar numericamente que algumas especialidades simplesmente não existem em determinadas regiões, como exemplo, temos a otorrinolaringologia com 0,18‰ do total de médicos, radiologia e diagnóstico por imagem com 0,29‰ e oftalmologia com 0,36‰ (SCHEFFER, 2013), com sua distribuição pelo país, por estado mostrado na Figura 26. É de grande interesse da comunidade médica brasileira o desenvolvimento de suporte tecnológico que

consiga minimizar os efeitos desta má distribuição e/ou que aumente a eficiência do atendimento médico.



PUC-Rio - Certificação Digital Nº 1521389/CA

Figura 26- Relação oftalmologista/população por estado Brasileiro. (CBO, 2014)

7.2

Apêndice B – Considerações Complementares Sobre a Abordagem, o Ato Médico e Seus Fundamentos Legais

Um sistema automatizado de apoio à priorização de atendimento é especialmente interessante no Brasil por ser um método restritamente em conformidade com a legislação em vigor neste país, conforme argumentação que se segue:

- a) Qualquer entidade que se proponha a gerar um sistema de exames diagnósticos (ou de diagnósticos associados a um exame) com aplicações práticas deverá, entre outras tarefas, se submeter à Metrologia Legal, e para isso, à aprovação do INMETRO, através de um processo de acreditação e submissão a ISO 17025 (ABNT NBR ISO/IEC 17025:2005), além de tangenciar uma área médico-legal que não está claramente definida em nossa legislação. A ISO 17025 é formada pela composição de 4 outras normas, conforme apresentado na Figura 27. A Lei 12.842/2013 (Lei do Ato Médico), estabelece definitivamente que o diagnóstico nosológico e o tratamento de doenças são competências restritas ao médico. As discussões sobre um diagnóstico automatizado ser ou não um ato médico não cessaram até o momento, cabendo então sua interpretação aos entes legais que porventura julguem a matéria.

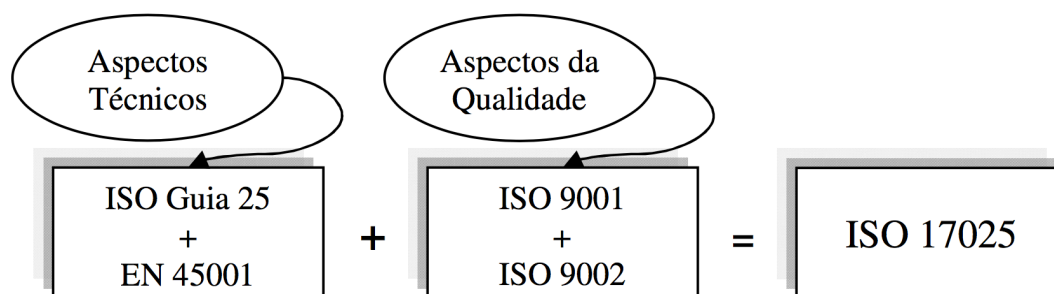


Figura 27 - Formação da ISO 17025

- b) Por outro lado, sistemas de apoio à priorização de pacientes já forma objetos de diversas consultas públicas, com parecer favorável ao

sistema. São exemplos as consultas públicas 16/2010 CRM-PR, 3463/2008 CRM-MG e se enquadram na resolução do Conselho Federal de Medicina (CFM) nº 1.627/2001 § 2º – *As atividades de prevenção primária e terciária que não impliquem na execução de procedimentos diagnósticos e terapêuticos podem ser atos profissionais compartilhados com outros profissionais da área da saúde, dentro dos limites impostos pela legislação pertinente.*

O termo **Apoio ao Diagnóstico**, utilizado por este trabalho é usado em seu sentido amplo, não apenas apoiar a identificação de patologias em exames, mas apoiar a identificação de indivíduos doentes.