UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ – UTFPR
DEPARTAMENTO ACADÊMICO DE COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

JHONATTAN SALVADOR GHELLERE

DETECÇÃO DE OBJETOS EM IMAGENS POR MEIO DA COMBINAÇÃO DE DESCRITORES LOCAIS E CLASSIFICADORES.

TRABALHO DE CONCLUSÃO DE CURSO

MEDIANEIRA
2015

JHONATTAN SALVADOR GHELLERE

DETECÇÃO DE OBJETOS EM IMAGENS POR MEIO DA COMBINAÇÃO DE DESCRITORES LOCAIS E CLASSIFICADORES.

Trabalho de Conclusão de Curso de Graduação do Curso Superior de Bacharelado em Ciência da Computação do Departamento Acadêmico de Computação – DACOM – da Universidade Tecnológica Federal do Paraná – UTFPR - Câmpus Medianeira, como requisito para obtenção do Título de Bacharel.

Orientadora: Profª. Alessandra Bortoletto Garbelotti Hoffmann, Msc.
Co-Orientador: Prof. Pedro Luiz de Paula Filho, Dr.

MEDIANEIRA

2015

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Departamento Acadêmico de Computação
Curso de Ciência da Computação

TERMO DE APROVAÇÃO

DETECÇÃO DE OBJETOS EM IMAGENS POR MEIO DA COMBINAÇÃO DE DESCRITORES LOCAIS E CLASSIFICADORES.

Por

JHONATTAN SALVADOR GHELLERE

Este Trabalho de Conclusão de Curso (TCC) foi apresentado às 15:50 h do dia 12 de junho de 2015 como requisito para a obtenção do título de Bacharel no Curso Superior de Ciência da Computação, da Universidade Tecnológica Federal do Paraná, *Campus* Medianeira. O acadêmico foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Profª. Msc. Alessandra Bortoletto Garbelotti Hoffmann
UTFPR – *Campus* Medianeira
(Orientadora)

Prof. Dr. Pedro Luiz de Paula Filho
UTFPR – *Campus* Medianeira
(Co-Orientador)

Prof. Dr. Arnaldo Cândido Júnior
UTFPR – *Campus* Medianeira
(Convidado)

Prof. MSc Juliano Rodrigo Lamb
UTFPR – *Campus* Medianeira
(Responsável pelas atividades de TCC)

A folha de aprovação assinada encontra-se na Coordenação do Curso.

AGRADECIMENTOS

Antes de tudo, a quem mais devo agradecer é, à minha mãe e minha avó, sem a educação, incentivo e esforço de vocês duas, provavelmente eu não teria oportunidade de estar finalizando este trabalho. Vocês são as duas pessoas mais importantes na minha vida.

Agradeço ao Professor Jean, primeiro por ter me dado a oportunidade de fazer parte de um grupo de pesquisa, e segundo por ter continuado me orientando mesmo após se desligar da universidade. Sem sua orientação e incentivo, o projeto e o desenvolvimento da pesquisa não teriam andado. Sou imensamente grato por cada minuto disponibilizado pelo senhor para ler e responder minha enorme quantidade de dúvidas.

Agradeço a Professora Alessandra, também por ter me dado oportunidade de fazer parte de um grupo de pesquisa e principalmente pelas contribuições e incentivos cruciais na finalização do trabalho. Também agradeço pela simpatia da professora durante o decorrer de todo o curso, precisamos sempre de professores assim!

Agradeço também ao Professor Pedro, por ter aceitado ser meu co-orientador sem pestanejar, fiquei muito honrado com essa atitude e agradeço enormemente pelas contribuições que forneceu para desenvolver o trabalho. Agradeço também pela gentileza que demonstrou durante todo curso com seus alunos, sempre disposto a ajuda-los, e também pelas oportunidades que me ofereceu.

Um agradecimento singelo ao Professor Juliano, por toda a disponibilidade para responder duvidas e pelas orientações dadas referentes a estrutura do projeto e aos meus colegas de turma do último período: Anderson, Everton e Felipe, todos contribuíram direta ou indiretamente para o trabalho.

Devo agradecer também à uma pessoa que ao certo não sei se realmente existe, já que às vezes penso que ela faz parte apenas da minha imaginação. De qualquer forma, sem o apoio das palavras dela, eu não teria conseguido ânimo para concluir este projeto ou até mesmo continuar o curso. Obrigado Daniela.

Por último, e não menos importante, tendo este trabalho sido realizado a partir do desenvolvimento das atividades do projeto de pesquisa, agradeço a Fundação Araucária pela concessão da bolsa de iniciação cientifica.

RESUMO

GHELLERE, Jhonattan Salvador. DETECÇÃO DE OBJETOS EM IMAGENS POR MEIO DA COMBINAÇÃO DE DESCRITORES LOCAIS E CLASSIFICADORES. Trabalho de Conclusão do Curso Superior de Ciência da Computação. Universidade Tecnológica Federal do Paraná. Medianeira, 2015, 90p.

A detecção de objetos em imagens permanece como um dos maiores desafios dentro da área de visão computacional, pois os objetos que estão contidos em imagens podem estar sob as mais variadas perspectivas e transformações de escala e rotação, o que torna mais complexa a meta de detectá-los. Esta tarefa possui aplicações nos mais diversos contextos, que vão desde o diagnóstico médico e área empresarial até ao que concerne à segurança pública. Com o objetivo de buscar uma solução para o problema de detecção de objetos genéricos, foi desenvolvido um módulo que se baseia na detecção via classificação. O módulo implementado, utiliza descritores locais para computar as características invariantes a transformações das imagens, o modelo Bag-of-Keypoints que se baseia em conceitos da área de recuperação de informação, para realizar a transformação dos dados extraídos das imagens, para serem compatíveis como entrada para aos indutores avaliados. Foram avaliadas as combinações de dois descritores locais (SIFT e SURF) com cinco abordagens de aprendizado supervisionado, sendo os algoritmos: *Multilayer Perceptron*, *FURIA*, *Random Forest*, *Support Vector Machines* e o *k-nearest neigbor*. A partir da construção do módulo e das análises dos resultados dos cenários experimentais e reais, verificou-se que os modelos gerados pelo módulo construído possuem altas taxas de acurácia nos cenários experimentais e resultados promissores no cenário real.

Palavras-chaves: detecção de objetos, *bag-of-keypoints,* redes sociais, aprendizagem de máquina.

ABSTRACT

GHELLERE, Jhonattan Salvador. OBJECT DETECTION, BAG-OF-KEYPOINTS, SOCIAL NETWORK, MACHINE LEARNING. Trabalho de Conclusão do Curso Superior de Ciência da Computação. Universidade Tecnológica Federal do Paraná. Medianeira, 2015, 90p.

The detection of objects in images remains one of the biggest challenges in computer vision area because the objects that are contained in images may be under the most varied perspectives and changes of scale and rotation, which makes more complex the goal to detect them. This task has applications in many different contexts, ranging from medical diagnosis and the business area until regards to public safety. In order to seek a solution to the problem of detection of generic objects, a module that is based on the detection via classification was developed. The implemented module uses local descriptors to compute the features invariant to transformations of images, the bag-of-Keypoints model based by concepts of area information retrieval, to perform the processing of data extracted from the images to be compatible as entrance to the evaluated inductors. Were evaluated, combinations of two local descriptors (SIFT and SURF) with five supervised learning approaches, the algorithms: Multilayer Perceptron, FURIA, Random Forest, Support Vector Machines and the k-nearest neigbor. From the module construction and analysis of experimental results and real scenarios, it was found that the models generated by the built module have high accuracy rates in experimental settings and promising results in real scenario.

Palavras-chaves: object detection, bag-of-keypoints, social network machine learning.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A visão computacional pode ser definida como um conjunto de técnicas e teorias para obtenção de informação por meio de imagens, sendo de modo geral, o estudo da capacidade das máquinas de visualizarem o mundo real (tridimensional), por meio de sensores, câmeras e outros dispositivos que extraem informação dos ambientes (ALVES et al., 2005).

Essa área é repleta de problemas com soluções parciais. Dentre estes problemas pode-se citar a detecção e o reconhecimento de objetos, o qual possuem inúmeras soluções, mas todas implementadas para resolução de situações especificas, não existindo uma solução que contemple uma resolução genérica para o reconhecimento de objetos.

Este problema no contexto da visão computacional, por exemplo, consiste em identificar a existência de objetos específicos em um cenário e, a partir desses objetos, reconhecê-los ou categorizá-los de acordo com uma necessidade qualquer, como por exemplo, identificá-lo em outras imagens.

Este problema pode ser resolvido de maneira específica, por meio do reconhecimento da forma do objeto desejado, como a identificação desta mesma forma no cenário de destino. Frequentemente essa técnica é chamada de *simple-matching* ou correspondência perfeita entre as imagens. No entanto, essa solução não permite o reconhecimento de outro objeto similar, por exemplo identificar a presença de uma face humana genérica em uma imagem a não ser que seja exatamente a face desejada.

Neste contexto, apresenta-se a intrínseca relação do processo de reconhecimento e a área de inteligência artificial. A aplicação das técnicas presentes nesta área, como aprendizado supervisionado, permite utilizar as informações obtidas pela visão computacional para fornecer ao sistema a capacidade de interpretação e aprendizado, e com isso, a possibilidade de se gerar conhecimento com base naquilo que foi ‘visto’; conhecimento este que pode inclusive ser usado em processos de tomada de decisão automática.

Com intuito de resolver o problema de detectar objetos genéricos em imagens, pretende-se com este projeto estudar e avaliar métodos de extração de características

a partir do processamento digital de imagens, bem como técnicas de inteligência computacional para a tarefa de classificação, e com isso encontrar uma combinação satisfatória dessas técnicas para solucionar este problema com base em imagens provenientes de redes sociais.

1.1 OBJETIVO GERAL

Identificar e aplicar técnicas de Inteligência Computacional e Processamento Digital de Imagens, para detecção de objetos genéricos em imagens provenientes de redes sociais.

1.2 OBJETIVOS ESPECÍFICOS

Para organizar o processo de desenvolvimento do trabalho, definiu-se os seguintes objetivos específicos:

a) Identificar os métodos de processamento digital de imagens para a detecção de pontos de interesse em imagens e, com base na literatura, selecionando o de maior relevância para a solução do problema proposto.
b) Avaliar os métodos de extração de características de imagens com ênfase nos que possuem invariância a transformações.
c) Identificar e selecionar bibliotecas de processamento de imagens e reconhecimento de padrões.
d) Avaliar os métodos de reconhecimento de padrões.
e) Analisar, determinar e avaliar uma abordagem para a tarefa de classificação das imagens extraídas considerando os paradigmas de Inteligência Artificial.
f) Analisar, projetar e desenvolver um componente responsável pelo reconhecimento dos objetos e classificação das imagens.

1.3 JUSTIFICATIVA

A comunicação atualmente feita por intermédio das redes sociais, não ocorre somente em texto, visto que o número de pessoas que acabam utilizando imagens para resumir pensamentos e até mesmo atos é crescentemente notório. Este fato está atrelado a necessidade natural do ser humano em relacionar-se, e que pode acabar levando-o a exposição pública de suas informações, gostos e opiniões seja por meio de textos como também imagens ou vídeos. Um exemplo dessa exposição se tem verificado com a popularização dos *selfies* (LAGAREIRO, 2014), que é o ato de fotografar a si mesmo estando ou não em companhia de outras pessoas.

Diversos estudos têm sido desenvolvidos (LECUN et al., 2014; LOWE, 2004; ALAHI et al., 2012, BAY et al., 2006) de modo a descobrir técnicas de reconhecimento e classificação de imagens, mesmo após elas terem sofrido algum tipo de transformação (escala e perspectiva por exemplo).

O processo de reconhecimento de objetos, entretanto, não é trivial. Segundo LECUN et al. (2014), o reconhecimento genérico com invariância a perspectiva, iluminação e a presença de ruídos é um dos maiores desafios da área de Visão Computacional.

Normalmente a correspondência entre duas imagens é feita usando uma cópia transformada, como por exemplo na detecção de semi-réplicas, que consiste na consulta das imagens que sofreram alguma transformação a partir da imagem original (BUENO, 2011). Com o presente trabalho procura-se detectar objetos em imagens a partir da classificação das imagens, como contendo ou não o objeto, ou seja, tendo como base de treinamento, imagens que contenham exemplos do objeto e imagens que não contenham o objeto.

Sendo assim, com a associação (por meio da geração e compartilhamento) de imagens aos mais diversos assuntos todos os dias, cria-se a necessidade de classifica-las (BOSCH et al., 2007). A partir das características extraídas, surge a importância prática de detectar objetos, que uma vez reconhecidos podem revelar posteriormente a descoberta de grupos de pessoas que compartilham mesmos interesses criando-se perfis ou até mesmo informações relevantes a segurança pública, alguns exemplos são a detecção de conteúdo impróprio como nudez (CARVALHO, 2012), objetos 3D dentro de bagagens por meio de imagens de raios-x (FLITTON et al., 2010)

e/ou de armas de fogo, o que pode possibilitar, posteriormente, a identificação do usuário associado a esse perfil.

2 PROCESSAMENTO DIGITAL DE IMAGENS E EXTRAÇÃO DE CARACTERÍSTICAS

Processamento Digital de Imagens (PDI) é uma área que se dedica ao estudo de teorias, modelos e algoritmos para a manipulação de imagens, feito por meio de computadores (BEZDEK et al., 2005), tendo origem, portanto, junto ao desenvolvimento dos mesmos. O primeiro registro de sua aplicação foi na utilização de técnicas para corrigir deformações no que foi a primeira imagem capturada da lua, ocorrida na década de 1960 com o desenvolvimento de programas espaciais norte-americanos (GONZALEZ e WOODS, 2002).

A partir deste ponto juntamente com a maior acesso aos computadores devido a diminuição do custo e aumento de desempenho do processamento, cresceu o número de áreas com aplicações, dentre elas:

- *Medicina e Biologia*: O PDI tornou possível a criação de diversos aparelhos que hoje são utilizados na medicina para aumentar a eficiência e rapidez do diagnóstico, como por exemplo a medicina nuclear, que utiliza a emissão coletada da detecção de raios gama para gerar imagens e localizar tumores em paciente (GONZALEZ e WOODS, 2002; ACHARYA e RAY, 2005; POLAKOWSKI et al., 1997; DOUGHERTY, 2009). Processar imagens de microscópio para contagem de células ou aumentar a precisão das tarefas de laboratório (DOUGHERTY, 2009; PEDRINI e SCHWARTZ, 2008) identificação de doenças no coração, realizar exames de mamografias digitais (ACHARYA e RAY, 2005). Classificação e analise do genoma (DOUGHERTY, 2009).
- *Astronomia, Sensoriamento Remoto e Meteorologia*: muito estudos estão sendo realizados na área de reconhecimento de padrões com aplicações na astronomia, como por exemplo, reconhecimento de supernovas por meio de aprendizagem de máquina (ROMANO et al., 2006), reconhecimento de objetos (TAGLIAFERRI et al., 1999), classificação de galáxias (DE LA CALLEJA e FUENTES, 2004; Ghellere et al., 2015), além disso, imagens astronômicas podem ser obtidas da emissão de raios gama de estrelas que explodiram a milhões de anos (STARCK e MURTAGH, 2006), imagens recebidas de satélites a partir do espaço podem ser processadas para análise

da situação ambiental e climática do planeta (ACHARYA e RAY, 2005) e também tornou possível realizar a detecção automática de tempestades solares (DOUGHERTY, 2009).

- *Segurança*: o uso de PDI na área de segurança é bastante útil para barrar o acesso não autorizado a locais ou recursos, e sua aplicação é evidente por meio do uso de reconhecedores de impressão digital, íris e até mesmo face, que servem como uma alternativa automatizada e até mesmo mais segura que os métodos tradicionais, para controlar acessos não autorizados, já que uma impressão digital é mais difícil de ser reproduzida que um cartão eletrônico (BOURIDANE, 2009).

Cada vez mais tem crescido o número de pesquisas na área para aumentar o desempenho na extração de características das imagens para posterior reconhecimento de padrões, porém, segundo GONZALEZ e WOODS (2002) não há consenso na literatura para estabelecer um limiar entre o PDI, Visão Computacional e a Inteligência Artificial. A relação entre quando uma começa e outra termina ou se, uma está contida na outra, ainda não é muito clara; Para tentar compreender melhor a relação entre as áreas, GONZALEZ e WOODS (2002) imaginaram uma linha contínua com o processamento digital de imagens em um extremo e a visão computacional no outro, dividida em três tipos de processos: processo nível baixo, médio e alto.

- *Processos de Nível Baixo*: Dentro desse nível se encontram as tarefas que vão desde a aquisição da imagem até filtragens, realces e outras tarefas de pré-processamento (GONZALEZ e WOODS, 2002). A aquisição da imagem depende muito do domínio do problema, elas podem ser geradas ou obtidas a partir de diversas fontes (i.e. raios gama, raios-x, sonar, ondas de rádio, micro-ondas, espectro visível, infravermelho, dentre outras) e meios (sensores, câmeras, satélites etc). Dentre as particularidades que devem ser levadas em conta nessa tarefa estão a escolha do meio de capturar a imagem, as cores, níveis de cinza e condições do ambiente (PEDRINI e SCHWARTZ, 2008; GONZALEZ e WOODS, 2002). Geralmente a aquisição de imagens a partir dessas fontes envolve alguma forma de pré-processamento, como por exemplo, redimensionar a imagem, aumentar ou diminuir o contraste para adequar a imagem à percepção da visão humana, visando corrigir imperfeições que podem ter ou não sido geradas pela etapa anterior

(PEDRINI e SCHWARTZ, 2008). O pré-processamento das imagens é responsável por operações primitivas para redução ou inclusão de ruídos, realce e restauração da imagem para que a mesma possa se enquadrar melhor no domínio do problema, tendo ao final apenas resultados com relação a qualidade da imagem (CONCI et al., 2008). Sendo assim, um processo de nível baixo é caracterizado por tanto a entrada quanto a saída do processo serem uma imagem (GONZALEZ e WOODS, 2002).

- *Processos de Nível Médio*: Neste nível estão tarefas como: segmentação, descrição e o reconhecimento. O processo de segmentação consiste em identificar regiões ou objetos, que sejam pertinentes ao domínio do problema e separá-los da imagem, com base em características como bordas e texturas, por exemplo (PEDRINI e SCHWARTZ, 2008). Realizar esta tarefa de forma autônoma, é um dos problemas mais difíceis dentro da área, pois ainda não há uma implementação genérica que realize a segmentação identificando objetos e regiões de forma a cobrir todos os que sejam de interesse, e esta tarefa é especialmente importante para se identificar objetos separados (GONZALEZ e WOODS, 2002). Extrair e descrever as características da imagem é próximo passo no processamento, regiões ou objetos que podem ter sido segmentados tem dados extraídos que possibilitam comparar ou diferenciar um objeto do outro, representados normalmente por vetores com atributos numéricos (PEDRINI e SCHWARTZ, 2008). Após a extração das informações que representam as características da imagem ou do objeto é realizado o reconhecimento. Esta etapa consiste em aplicar um tipo ao objeto com base nos dados extraídos de modo a dar um significado ao mesmo, como por exemplo aplicar o rótulo “animal” ou “avião” (PEDRINI e SCHWARTZ, 2008; GONZALEZ e WOODS, 2002). Os processos de nível médio são caracterizados por entradas serem imagens mas as saídas serem dados (GONZALEZ e WOODS, 2002).
- *Processos de Nível Alto*: Grande parte das pesquisas de visão computacional são voltadas para problemas “*o que”* e “*onde”*, como por exemplo detecção e classificação de objetos, porém é possível pensar além dessas tarefas, como classificar imagem se baseando em conceitos abstratos, como pacificidade (CIPOLLA et al., 2012), tarefas como esta estão ligadas com a área de cognição e de Inteligência Artificial. Elas buscam dar sentido ao que

a máquina vê, ao mesmo tempo que tentam emular a inteligência humana, de modo a possibilitar que a mesma aprenda, faça inferências e tire conclusões com base no que está enxergando no ambiente (GONZALEZ e WOODS, 2002).

Uma imagem digital, formada por um único canal de cor, pode ser definida como uma função bidimensional $f(x, y)$ sendo que (x, y) indica a coordenada espacial pertencente à $\mathbb{R}^2$ que representa a posição de um ponto, e o valor de f representa a intensidade ou nível de cinza do ponto na imagem (BALLARD e BROWN, 1982). Uma imagem que possui mais de um canal de cor terá uma função representando cada canal de cor, como por exemplo o padrão BGR (*Blue, Green, Red*) (LIBERMAN, 1997; BALLARD e BROWN, 1982), que é esquematizado na Figura 1. Nota-se que, um ponto (x, y) no espaço da imagem é representado por vetor no qual cada componente é um valor escalar representando respectivamente a intensidade de cada cor.

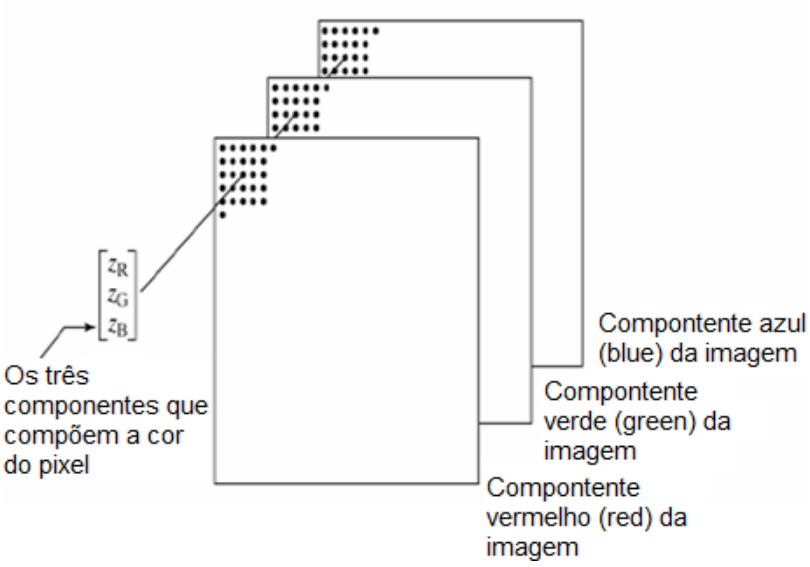


Figura 1 - Imagem formada por três canais de cores.
Fonte: Paula Filho (2014)

A representação da imagem pode ser feita computacionalmente por meio de uma matriz $M(i, j)$ formada por um número de elementos finitos denominados *pixels*, que possuem um valor escalar e uma localização (i, j) a partir da origem da imagem.

Cada *pixel* possui uma relação com seu vizinho. Segundo Gonzalez e Woods (2002) cada pixel p possui quatro vizinhos horizontais e verticais representados pelas coordenadas:

$$(x+1, y), (x-1, y), (x, y+1), (x, y-1) \quad (1)$$

Esse conjunto de pontos é também chamado de *vizinhança-4* de p, e pode ser expresso por $N_4(p)$.

Há também o conjunto de vizinhos diagonais de p, expresso por $N_D(p)$ e é representado pelas coordenadas:

$$(x+1, y+1), (x+1, y-1), (x-1, y-1), (x-1, y-1) \quad (2)$$

Seguindo essa ideia, há também a chamada vizinhança-8 de p, descrita por (PEDRINI e SCHWARTZ, 2008) como:

$$N_8(p) = N_4(p) \cup N_D(p)$$

Onde $N_8(p)$ contém todas as coordenadas de $N_4(p)$ e $N_D(p)$, a não ser que o pixel p esteja localizado na borda da imagem, neste caso algumas coordenadas não aparecerão no conjunto (GONZALEZ e WOODS, 2002). Por meio da Figura 2, são apresentados os modelos de vizinhança, sendo (a) a vizinhança-4 de p, (b) vizinhança-D de p e (c) a vizinhança-8 de p.

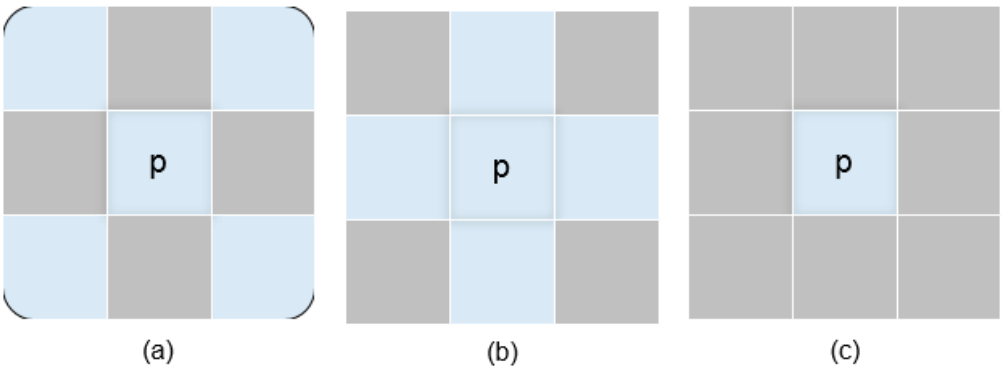


Figura 2 - Vizinhanças de um *pixel p.*
Fonte: Adaptado de GONZALEZ e WOODS (2002).

2.1 CARACTERÍSTICAS DA IMAGEM

O método mais conhecido para procurar por imagens é compará-las diretamente. Isso é feito com a correspondência dos valores de um pixel ou um conjunto de pixels (região) de uma imagem com os valores de uma outra imagem (DESELAERS, 2003).

Esses conjuntos de pixels uma vez analisados podem revelar as características de uma imagem, que são propriedades cujos valores devem ser semelhantes para os objetos em uma classe particular, e diferentes para os objetos em outra classe ou mesmo em relação ao fundo da imagem (DOUGHERTY, 2013).

Portanto, para que o computador consiga processar os dados de forma a reconhecer semanticamente o que está presente na imagem, o primeiro passo é detectar e extrair características de forma eficiente e efetiva (TIAN, 2013). Entre os exemplos dos detalhes e características que os humanos conseguem extrair de uma imagem com percepção visual estão a identificação de picos em montanhas, cantos e portas de construções, caminhos através da neve, (SZELISKI, 2010), além de uma região de textura, como pode ser visto por meio da Figura 3.

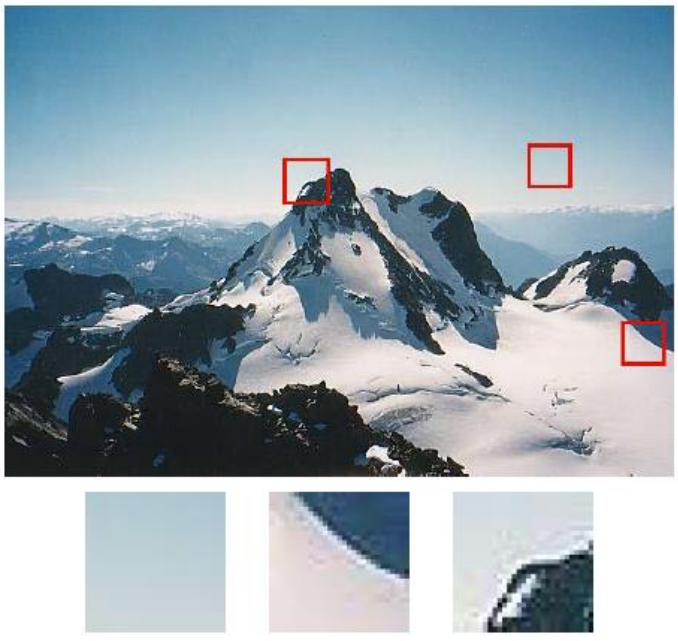


Figura 3 - Exemplos de caracteristicas
Fonte: SZELISKI (2010)

A escolha das características corretas da imagem depende muito do domínio do problema ou da própria imagem, mas geralmente procura-se por características que sejam invariantes a possíveis transformações que a imagem possa sofrer, como translação, mudança no tamanho e rotação (DOUGHERTY, 2013; NIXON e AGUADO, 2002).

As características de uma imagem podem ser extraídas a partir de informações locais e/ou globais. Características globais tem a propriedade de generalizar um objeto inteiro com apenas um vetor (LISIN et al., 2005) e geralmente algoritmos que extraem informações globais tendem a ser mais rápidos e simples (PENATTI, 2009) e com boa tolerância a ruídos (DE ARAÚJO, 2009). Alguns exemplos de características globais extraídas de uma imagem pode ser: cor (RAOUI et al., 2011; PAULA FILHO, 2013) geralmente extraídas por meio de um histograma (PENATTI, 2009), textura (PAULA FILHO, 2013; LIBERMAN, 1997) e forma (DESELAERS, 2003; LISIN et al., 2005; RAOUI et al., 2011).

Contudo, características globais localizam poucos detalhes e são ineficientes quando a imagem sofre alguma alteração em sua perspectiva, como por exemplo, ser rotacionada, transladada, e ao mesmo tempo são sensíveis a oclusões, desordens ou deformações no objeto (GRAUMAN e LEIBE, 2011; DE ARAÚJO, 2009; LISIN et al., 2005; DE ANDRADE, 2012).

Uma característica local, que é o foco deste trabalho, é descrita como um padrão dentro da imagem que difere de sua vizinhança, associada a mudança de uma ou várias propriedades da mesma, como intensidade, cor ou textura, podendo ser representadas por pontos de interesse, bordas ou trechos da imagem (TUYTELAARS e MIKOLAJCZYK, 2008).

Pontos de interesse são pontos que possuem uma posição bem definida dentro da imagem e cuja região a sua volta é rica em informação, tornando-o um ponto único, ou seja, que não se repete e distingue-se dos demais. Estes pontos, também conhecidos como pontos chaves, segundo (SCHMID et al., 1998), são comumente localizações na imagem onde o sinal muda em duas dimensões, ou ainda onde há a intersecção de duas bordas ou onde a própria borda da imagem muda subitamente.

Para (TUYTELAARS e MIKOLAJCZYK, 2008) características locais ideais devem ter as seguintes propriedades:

- *Repetição*: Dadas duas imagens do mesmo objeto ou cena, retiradas de condições diferentes de visualização, uma elevada percentagem das características detectadas na parte visível de ambas as imagens, devem ser encontradas em ambas as imagens.
- *Distinção* / *Informação*: Os padrões de intensidade subjacentes às características detectadas devem mostrar muita variação, de tal forma que as características possam ser distinguidas e correspondidas.
- *Local*: As características devem ser locais, de modo a reduzir a probabilidade de oclusão e permitir simples aproximações de modelo das deformações geométricas e fotométricas entre duas imagens obtidas sob diferentes condições de visualização.
- *Quantidade*: O número de características detectadas deve ser suficientemente grande, de tal modo que um número razoável de características possa ser detectado mesmo em pequenos objetos. No entanto, o número ótimo de características depende da aplicação. A densidade de características deve refletir o conteúdo de informação da imagem para fornecer uma representação compacta da imagem.
- *Precisão*: As características detectadas devem ser localizadas com precisão, tanto no local da imagem, quanto no que diz respeito à escala e forma.
- *Eficiência*: A detecção de características em uma nova imagem deve ser eficiente computacionalmente considerando a aplicação.

Ainda Segundo TUYTELAARS e MIKOLAJCZYK (2008) a repetição é a propriedade mais importante entre as citadas e pode ser alcançada de duas maneiras: invariância e robustez.

- *Invariância*: Quando são esperadas grandes deformações, a abordagem preferencial é modelar matematicamente essas deformações, se possível, e, em seguida, desenvolver métodos para detecção de características que não são afetados por essas transformações matemáticas.
- *Robustez*: No caso de pequenas deformações, que muitas vezes é suficiente para tornar os métodos de detecção de características menos sensível a tais deformações, isto é, a precisão da detecção pode diminuir, mas não tão drasticamente. Deformações típicas que são abordadas utilizando robustez são o ruído de imagem artefatos de compressão, borrão, etc.

Tais características podem ser usadas para encontrar conjuntos esparsos de correspondências locais em diferentes imagens (SZELISKI, 2010) e portanto formam a base para as abordagens de reconhecimento de objetos específicos ou uma classe de objetos (GRAUMAN e LEIBE, 2011).

2.2 DESCRITORES DA IMAGEM

Após a detecção das características é preciso fazer a correspondência, ou seja, saber quais características detectadas são encontradas em outras imagens (SZELISKI, 2010).

Mas para realizar a correspondência para classificação a partir do reconhecimento de um objeto (ou vários) é preciso descrever as características detectadas (extraídas) do conjunto de pixels (ou pontos de interesse), geralmente representadas por um conjunto de números (vetor) que representam as características do objeto, de modo que seja possível classificar uma imagem (NIXON e AGUADO, 2002).

Um exemplo é a *Correspondência Perfeita* que é demonstrado na Figura 4, na qual é comparado através de uma medida de similaridade os descritores extraídos a partir das características detectadas de duas imagens, tendo como base uma Imagem *Template* que representa o objeto de interesse que se procura detectar nas imagens a serem analisadas.

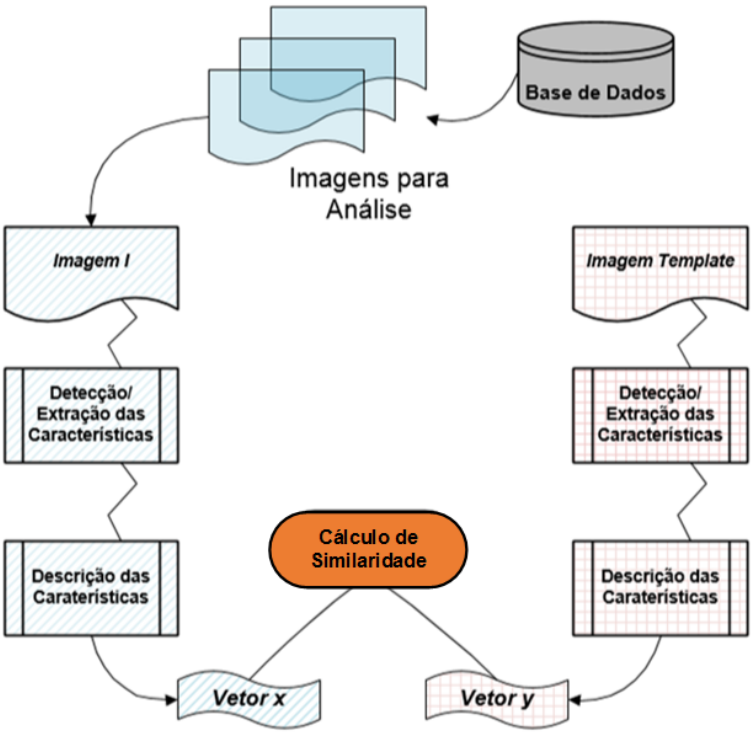


Figura 4 - Exemplo de *Correspondência Perfeita*.
Fonte: Autoria Própria.

Descritores são muito aplicados na área de CBIR *(Content-based image retrieval*) que é a tarefa de procurar imagens em um banco de imagens (ALMEIDA, TORRES e GOLDENSTEIN, 2009; BUENO, 2011). A consulta a base de dados pode ser de vários tipos, por texto onde o usuário faz uma descrição textual da imagem que ele está procurando, por esboço, onde o usuário fornece um esboço da imagem que ele está procurando e por fim por exemplo, onde o usuário dá uma imagem de exemplo semelhante ao que ele está procurando (DESELAERS, 2003).

Nos dois últimos casos descritores podem ser aplicados para realizar o casamento entre as imagens por meio de uma busca no banco de dados onde são retornadas as imagens mais similares à imagem ou segmento de busca, processo resumidamente representando na Figura 5.

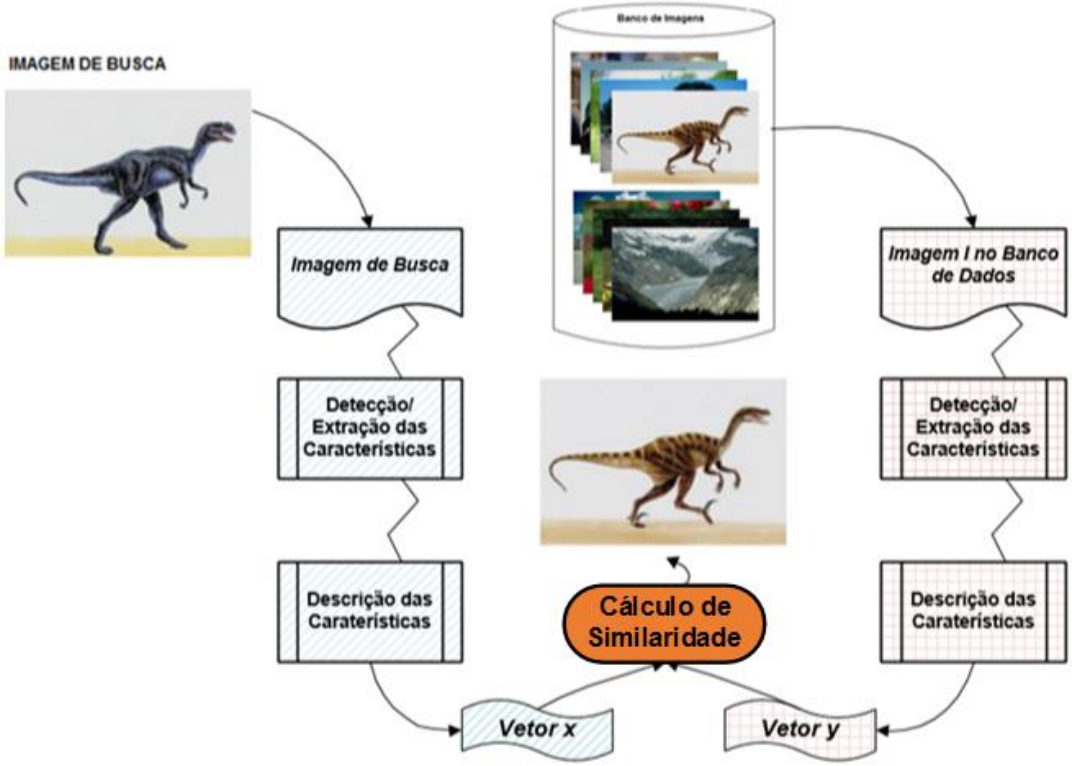


Figura 5 - Exemplo de CBIR
Fonte: Adaptado de DESELAERS (2003)

2.2.1 Scale Invariant Feature Transforms – SIFT

O algoritmo SIFT (*Scale Invariant Feature Transforms*) foi proposto por Lowe para a extração de características que fossem invariantes, de modo a possibilitar a correspondência de objetos dentro de cenas mesmo sobre diferentes pontos de perspectiva, após terem sofrido alguma transformação, seja na escala, rotação ou iluminação (LOWE, 2004). A característica de invariância tornou o algoritmo bastante reconhecido, tendo bastante aplicabilidade em diversos contextos de visão computacional e aprendizagem de máquina, como por exemplo, criação de imagens panorâmicas através da junção de várias imagens (BROWN e LOWE, 2007), mapeamento e navegação de robôs, identificação de digitais (PARK et al., 2008), reconhecimento de objetos (LOWE, 1999; SUGA, 2008), reconhecimento de objetos 3D (FLITTON, 2010) dentre outros.

A obtenção de descritores é feita segundo Lowe (2004) em quatro etapas, conforme a Figura 6, e serão descritas brevemente a seguir.

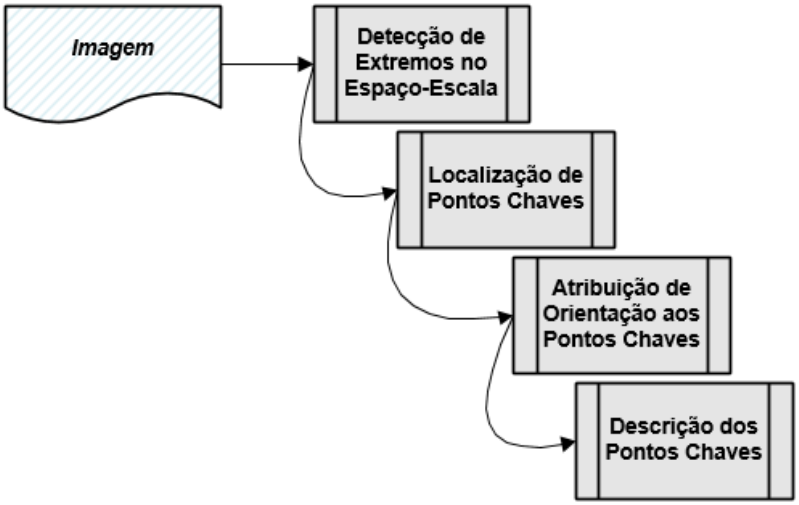


Figura 6 - Etapas para obteção do descritor SIFT.
Fonte: Autoria Própria.

A detecção de extremos no espaço-escala é realizada identificando potenciais candidatos a pontos de interesse, e de acordo com Lowe (2004) procurando extremos (valores locais máximos e mínimos), por meio da aplicação da função de diferença Gaussiana num espaço de escala de uma pirâmide de imagem, que é construída por meio das chamadas oitavas (*octaves*). A cada oitava a imagem inicial se torna a imagem cujo valor σ (sigma, fator de Filtro Gaussiano) é o dobro do inicial, ao mesmo tempo a imagem tem sua escala reduzida pela metade.

A aplicação da função de Diferença Gaussiana, como propôs (LOWE, 2004) consiste em aplicar Filtros Gaussianos que borram a imagem por meio de um fator σ (sigma) que varia em um intervalo na imagem e em seguida subtrai-la de outra com um valor σ diferente. Na Figura 7 é possível ver o esquema exemplificando a aplicação das diferenças gaussianas, a cada oitava (*octave*) é aplicado o Filtro Gaussiano com valor σ, em seguida é feita a subtração das imagens que resultam em Diferenças de Gaussiana, a próxima oitava inicia com a imagem que possui o dobro de σ inicial.

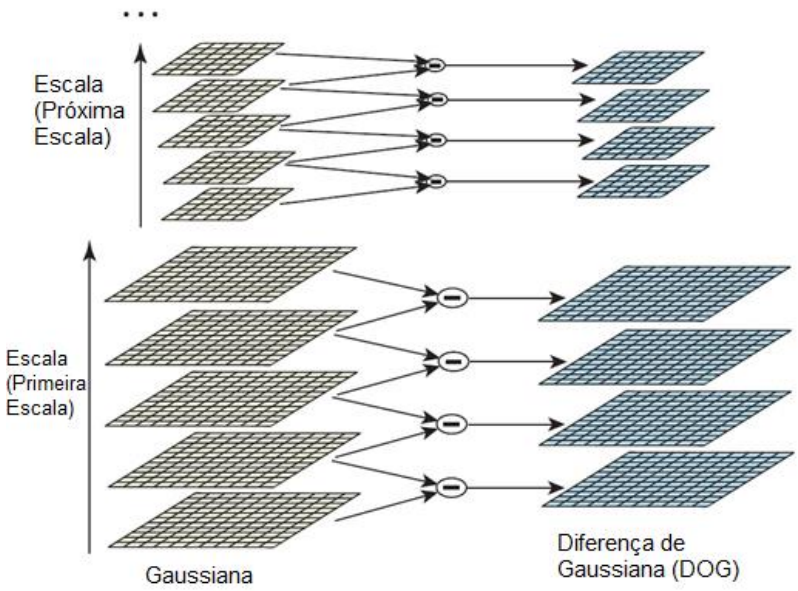


Figura 7 - Exemplo da função *DoG* para cada oitava.
Fonte: Lowe (2004).

Após a aplicação da Diferença Gaussiana para cada oitava, é realizado a verificação para encontrar extremos. Essa busca é feita para cada intervalo de diferença gaussiana (excluindo a imagem com fator σ mínimo e máximo da oitava), como pode ser visto na Figura 8 (BELO, 2006), onde a busca por extremos é realizada apenas nas imagens dentro do intervalo em vermelho.

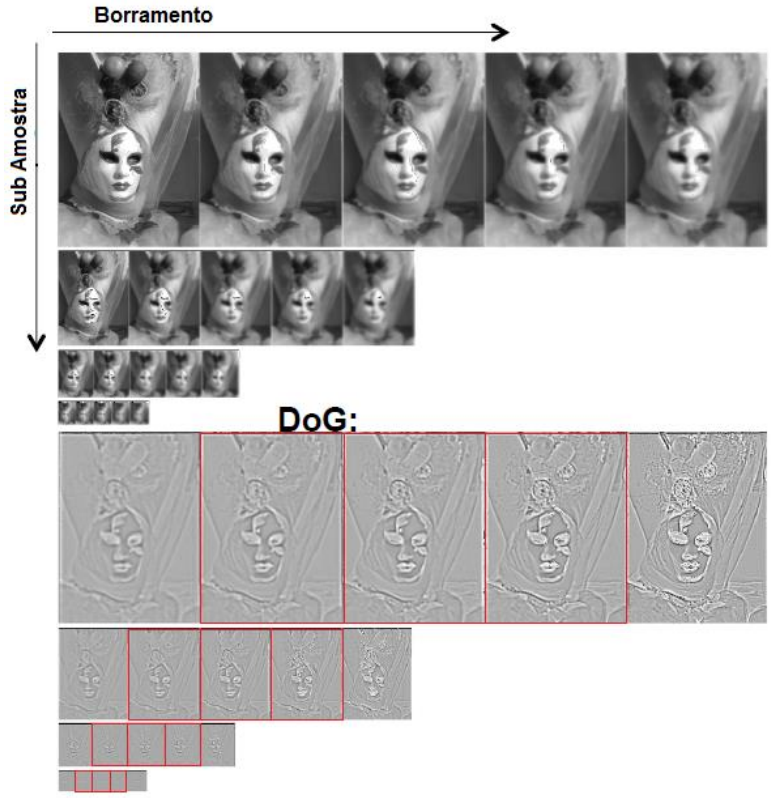


Figura 8 - Exemplo da aplicação DoG em cada oitava.
Fonte: Adaptado de LEUTENEGGER (2012)

Para se obter os possíveis candidatos a pontos chaves é feita a análise de cada pixel com seus 26 vizinhos nas diferentes escalas de sigma. A Figura 9 mostra o pixel X marcado, caso ele tenha valor maior ou menor que todos os seus vizinhos será um possível ponto de interesse (Lowe, 2004).

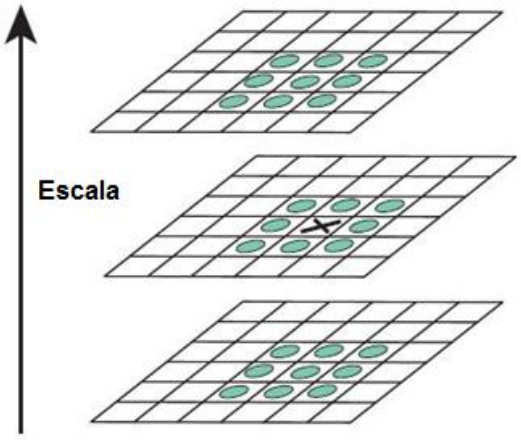


Figura 9 - Detecção de extremos no espaço escala dos intervalos.
Fonte: Lowe (2004)

A localização dos candidatos a pontos chave utilizando a detecção de extremos gera muitos pontos que por sua vez são instáveis. Para determinar com maior precisão a localização e estabilidade dos pontos, Lowe (2004) propõe a utilização do cálculo de localização interpolada por meio da expansão quadrática da série de Taylor sobre a função da Diferença Gaussiana, e em seguida são filtrados os pontos por meio de um limiar de contraste, para descartar pontos com valores inferiores ao mesmo que é feita por meio do cálculo da expansão da série de Taylor de segunda ordem da Função de Diferença Gaussiana.

De modo a otimizar e selecionar pontos mais estáveis, é necessário remover os que possuem má localização, e que possuem forte resposta a arestas devido a função DoG (diferenças gaussianas). Isto é feito utilizando o Hessiano (Matriz hessiana) com as derivadas parciais de segunda ordem da função DoG (Lowe, 2004).

Atribuição de orientação aos pontos chave é realizada para obter a invariância a rotação, Lowe, (2004) propôs o cálculo de orientações e magnitudes do gradiente para cada ponto chave, com base nas diferenças dos pixels pertencentes a imagens presentes nos intervalos do espaço-escala. Um histograma é montado representando todas orientações entre 0 e 2π e cada ponto vizinho ao ponto chave é adicionado ao histograma. O pico do histograma servirá para definir a orientação do ponto chave, mas além dele, picos que possuem ao menos 80% o seu valor, também serão considerados, o que poderá levar o ponto chave a ter mais de uma orientação (Lowe, 2004).

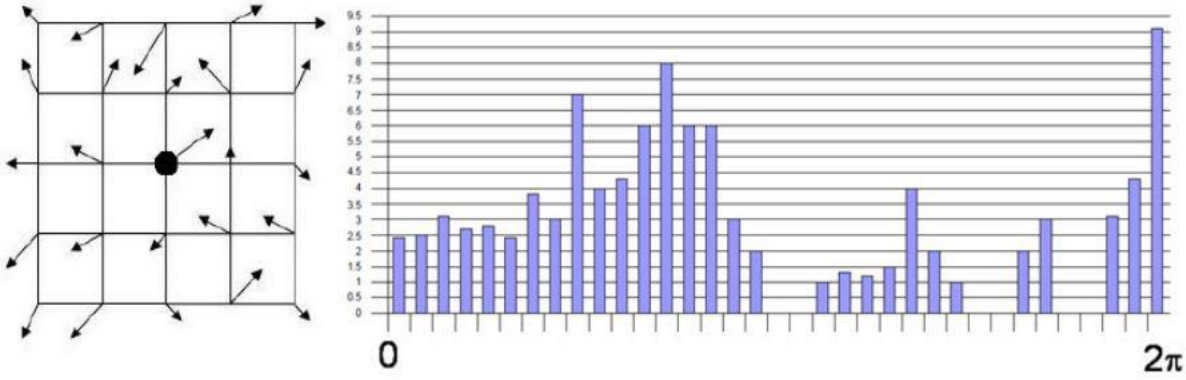


Figura 10 - Histograma dos gradientes.
Fonte: Adaptado de Gonzáles & Meggiolaro (2010)

A construção do descritor é feita considerando uma região de tamanho 4 ao redor do ponto chave dividida em sub-regiões 4x4. Para cada sub-região é calculado

um histograma considerando 8 orientações, o que leva a criação de um vetor de tamanho 128 que descreve o ponto chave, computado a partir das magnitudes e orientações dos gradientes (Lowe, 2004).

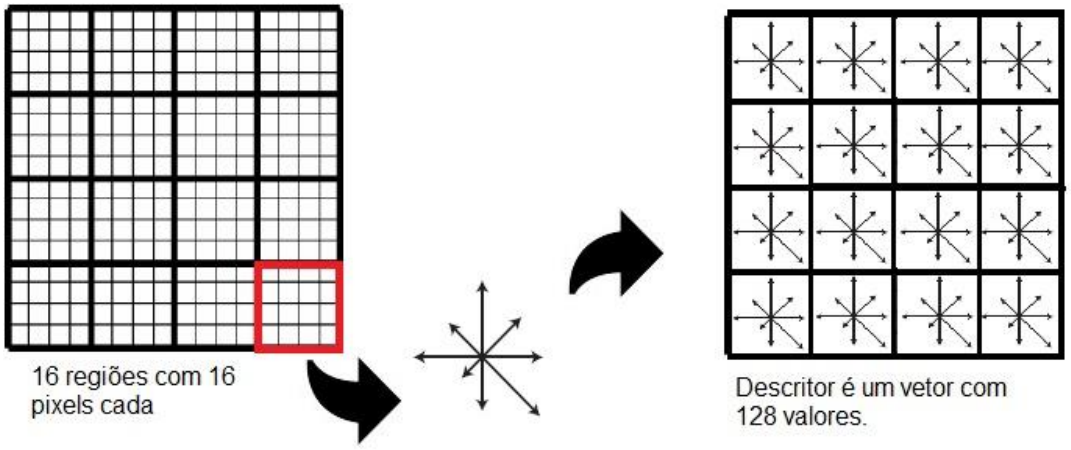


Figura 11 - Exemplo de Histograma para cada uma das 16 regiões.
Fonte: Adaptado de Belo (2006)

A função gaussiana é aplicada na região à volta do ponto chave para dar peso a magnitude do gradiente de cada ponto na vizinhança para evitar mudanças súbitas na posição e dar menos ênfase a gradientes que estão longe do ponto chave (GONZÁLES e MEGGIOLARO, 2010). Na Figura 12 é demonstrado a gaussiana como um círculo azul em volta da região.

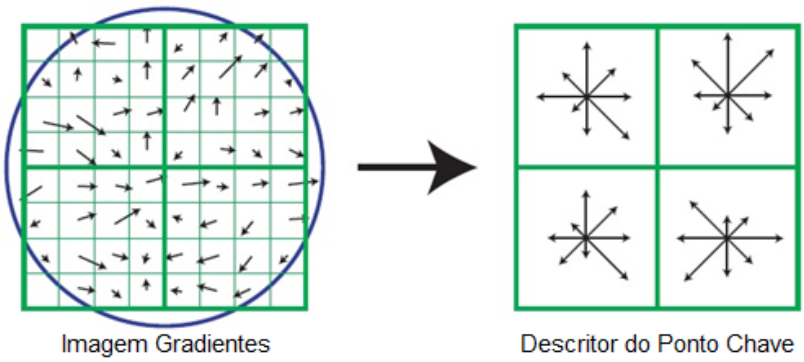


Figura 12 - Região de aplicação da Gaussiana para enfatizar os pontos vizinhos.
Fonte: Lowe (2004)

2.2.2 Speeded-Up Robust Features – SURF

O SURF (*Speeded-Up Robust Features*) foi proposto em 2006 por (BAY et al., 2008), e tem uma abordagem muito similar ao SIFT, porém, encontra os pontos de interesse e os descreve computacionalmente mais rápido (BUENO, 2011).

Para fazer a detecção de pontos de interesse, diferentemente do SIFT que usa Diferenças Gaussianas, o SURF realiza esse processo por meio de filtros Haar (uma forma de calcular wavelet) que tem formato de caixa e se baseia no determinante da Matriz Hessiana computando as derivadas de segunda ordem da Gaussiana, utilizando integrais de imagens que permitem a rápida computação dos filtros (Figura 13) (BAY et al., 2008), para obter a localização e escala do ponto chave, as localizações são interpoladas e procura-se descartar pontos de baixo contraste ou localizados em arestas, similarmente ao que é feito no SIFT.

O espaço escala também é analisado iterativamente aumentando a escala do filtro e reduzindo o tamanho da imagem (BARBOSA, 2014).

Para fazer a atribuição, Bay et al. (2008) propuseram a realização de atribuição de orientação utilizando as respostas da aplicação de *wavelet* na direção horizontal e vertical para a vizinhança do ponto.

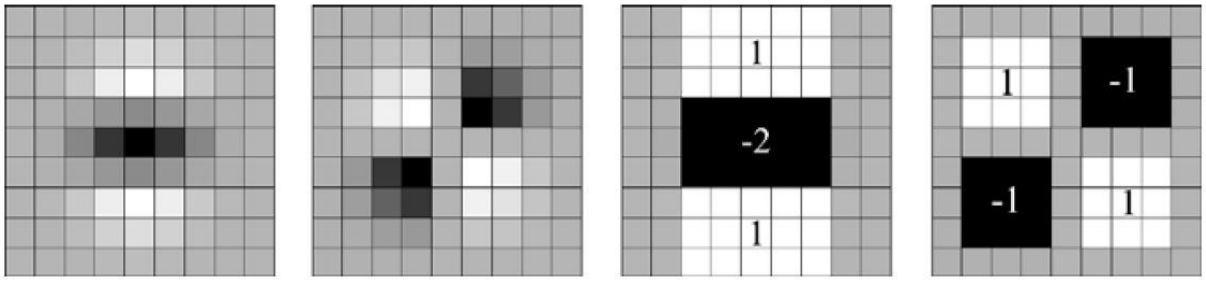


Figura 13 - Aplicação das derivadas de 2ª ordem da Gaussiana e do Filtro Caixa.
Fonte: BAY et al. (2008).

O descritor é computado com base na construção de um histograma de orientação dos gradientes dos pontos presentes nas 16 regiões ao redor do ponto de interesse. Para cada sub-região é analisada a soma das respostas dos filtros e o modulo das respostas nas direções vertical e horizontal (Figura 14), ou seja, 4 valores, uma

para cada direção é computado, para cada sub-região, resultando em um vetor de 64 dimensões (BAY et al., 2008).

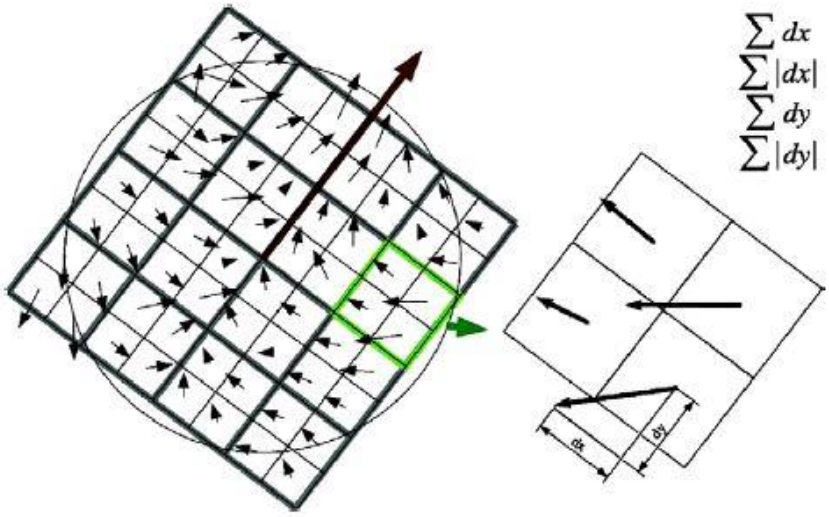


Figura 14 - Descritor SURF.
Fonte: BAY et al. (2008).

3 RECONHECIMENTO DE PADRÕES EM IMAGENS

O Reconhecimento de Padrões pode ser entendido como uma ampla área de estudo que tem a ver com tomadas de decisões automáticas (STARCK e MURTAGH, 2006) ou ainda uma área de estudo científico que tem como escopo a classificação de objetos em um número de categorias ou classes (THEODORIDIS e KOUTROUMBAS, 2006).

É ainda uma das etapas do processo de descoberta de conhecimento (*Knowledge Discovery Process*, KDP), que pode ser definido como um processo não trivial de extração de informação desconhecida e útil dos dados (CIOS *et al.*, 1998) resumido graficamente por meio da Figura 15.

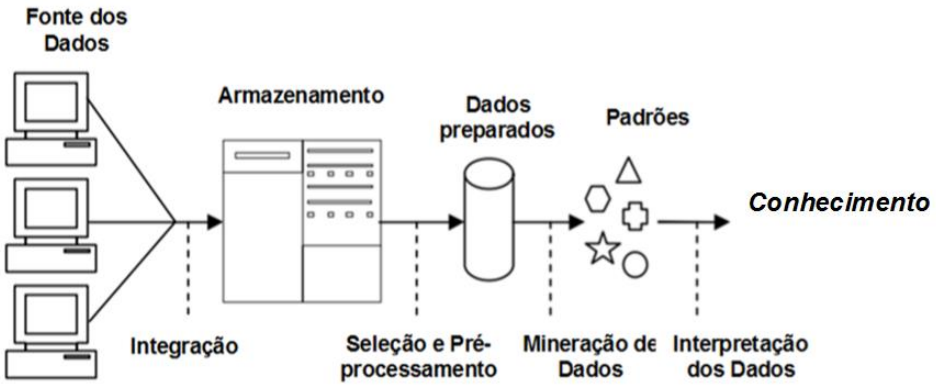


Figura 15 - Processo de Descoberta de Conhecimento.
Fonte: adaptado de BRAMER (2007)

Os dados são obtidos de diversas fontes e são integrados e armazenados em algum local. Os mesmos são selecionados e pré-processados sobre os quais posteriormente são utilizados algoritmos de mineração de dados para a descoberta de padrões que são interpretados por um especialista do domínio do problema, produzindo conhecimento (BRAMER, 2007).

Em imagens, um padrão pode ser definido como um conjunto de descritores, que são as características extraídas da imagem (GONZALEZ e WOODS, 2002) e estes padrões podem ou não caracterizar objetos. Portanto, de maneira semelhante é constituído o processo para reconhecimento de padrões em imagens. Como abordado no Capítulo 2 as imagens podem vir de diversas fontes, sobre elas são realizadas

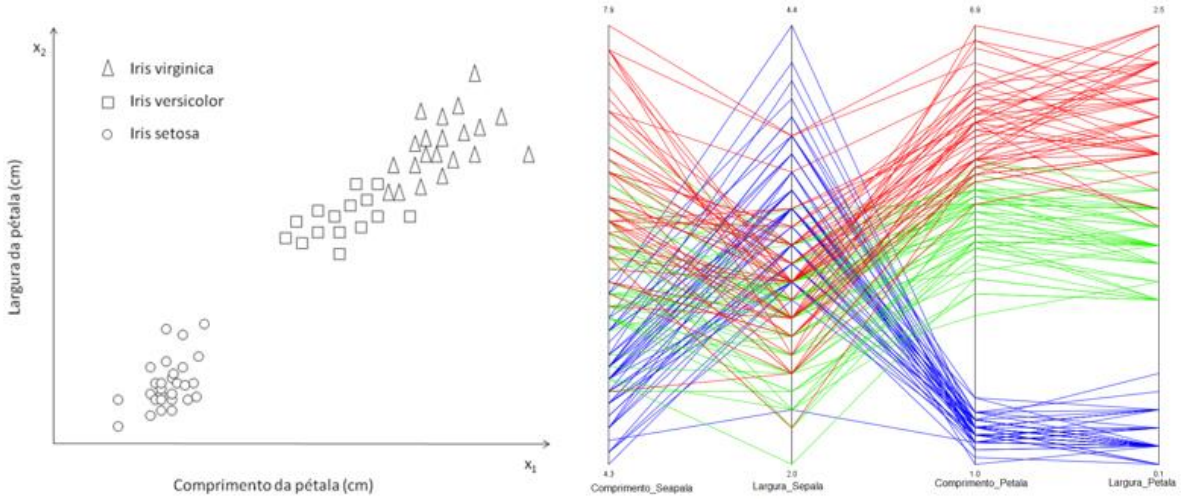
tarefas de pré-processamento, suas características são extraídas e um algoritmo de aprendizado é utilizado para aprender esses padrões e reconhecê-los em outras imagens.

O reconhecimento é o problema central da aprendizagem de categorias visuais e, em seguida vem a questão de identificar novas ocorrências dessas categorias. O próprio reconhecimento Visual tem uma variedade de aplicações potenciais que tocam muitas áreas da inteligência artificial e recuperação de informação, incluindo, por exemplo, o CBIR (*Content Based Image Retrieval*) (GRAUMAN e LEIBE, 2011).

Todos os dias as pessoas reconhecem rostos em torno delas, porém, e isso é feito de maneira inconsciente e por não haver um modo de conseguir explicar essa experiência, tem-se a dificuldade de escrever um programa de computador para que faça o mesmo. Por exemplo, cada face humana tem um padrão composto de combinações de estruturas particulares (olhos, nariz, boca etc.) localizadas em posições definidas na mesma. Pela análise de amostras de imagens de faces, um programa deve ser capaz de extrair um padrão específico para um rosto e identificá-la ou reconhecê-la como sendo um rosto, isso seria reconhecimento de padrões (DOUGHERTY, 2013).

O reconhecimento de padrões pode ser também observado por meio de diagramas e gráficos, onde se tem uma visão mais inteligível dos dados. Como exemplo considere o conjunto Íris[1], que possui exemplos de 3 espécies de plantas, com cerca de 50 exemplos de cada espécie com 4 atributos especificados (Comprimento e Largura da sépala, Comprimento e Largura da pétala), na Figura 16 está esboçado as três espécies presentes no mesmo. É possível observar pela Figura 16a, a distribuição de exemplos de cada espécie segundo os valores referentes a dois atributos (características) das flores. No caso da espécie *Setosa* observa-se que a mesma se separa consideravelmente das outras duas, este é um padrão que pode ser observado graficamente. Na Figura 16b observa-se a separação das espécies por meio dos valores dos quatros atributos plotados no gráfico de coordenadas paralelas, por meio dele, é possível observar atributos que separam melhor as espécies, no caso os dois atributos que formam os eixos do gráfico plotado na Figura 16a.

[1] Disponível em https://archive.ics.uci.edu/ml/datasets/Iris



(a) – Conjunto Íris representado no plano Cartesiano

(b) – Conjunto Íris representado por coordenadas paralelas

Figura 16 - Visualização de padrões no Conjunto Íris.
Fonte: (a) Adaptado de GONZALEZ e WOODS (2002) (b) Autoria Própria.[2]

Nas próximas sessões serão descritos brevemente conceitos essenciais para o entendimento do processo de reconhecimento de padrões em imagem, visando a detecção de objetos.

3.1 INSTÂNCIAS E CLASSES GENÉRICAS DE OBJETOS

O mundo real é constituído de um amontoado de objetos, e eles podem aparecer um na frente do outro e em diferentes perspectivas (SZELISKI, 2010) o que representa um grande desafio na tarefa de reconhecimento de objetos, para tanto, no Capitulo 2 foram apresentados algoritmos que se utilizam de características locais para ser possível encontrá-las mesmo após a imagem ter sofrido algum tipo de transformação ou oclusão.

Contudo, a extração das características é só uma parte do processo, para se chegar ao reconhecimento, sendo este geralmente considerado por pesquisadores da visão computacional como sendo de dois tipos: o caso específico e a de classe genérica. No caso específico, procura-se identificar as instâncias de um determinado objeto, lugar ou exemplo de pessoa (GRAUMAN e LEIBE, 2011).

[2] Visualização gerada a partir da ferramenta Weka (http://www.cs.waikato.ac.nz/ml/weka/)

O segundo tipo é descrito pela literatura como sendo o tipo de reconhecimento mais desafiador, que é o reconhecimento de categoria genérica (ou classe), que pode envolver reconhecer instâncias de classes extremamente variadas, tais como animais ou móveis, (SZELISKI, 2010). Neste nível, busca-se reconhecer diferentes instâncias de uma categoria genérica como pertencendo ao mesmo exemplo conceitual da classe (GRAUMAN e LEIBE, 2011).

Na Figura 17 são apresentados exemplos de objetos específicos e genéricos.

Figura 17 - Objetos genéricos vs Objetos Específicos
Fonte: Adaptado de GRAUMAN e LEIBE (2011)

Os paradigmas principais para o reconhecimento de objeto específico se baseiam na correspondência (*Matching*) e na verificação geométrica. Já, o reconhecimento genérico de objetos, inclui modelo estatístico de sua aparência ou aprendizado de sua forma a partir de exemplos (GRAUMAN e LEIBE, 2011).

3.2 APRENDIZAGEM DE MÁQUINA

O Aprendizado de máquina é uma área interdisciplinar de pesquisa, usualmente, se referindo as mudanças em sistemas que executam tarefas associadas a Inteligência Artificial, tais como: reconhecimento, diagnóstico, planejamento, controle de robôs, predições, etc (NILSSON, 1996), executadas por meio da construção de programas que são capazes de se aperfeiçoar com experiência (Mitchell, 1997).

A aprendizagem de máquina se vale do raciocínio indutivo para realizar a descoberta de padrões ou generalizar a respeito de um certo conjunto de dados, com o intuito de tirar alguma conclusão. Esse tipo de raciocínio é feito sobre exemplos que são fornecidos ao algoritmo, esse modelo de aprendizado se divide em duas categorias, aprendizagem supervisionada e aprendizagem não supervisionada (MONARD e BARANAUSKAS, 2003), um esquema demonstrando a ramificação do aprendizado indutivo pode ser visto na Figura 18.

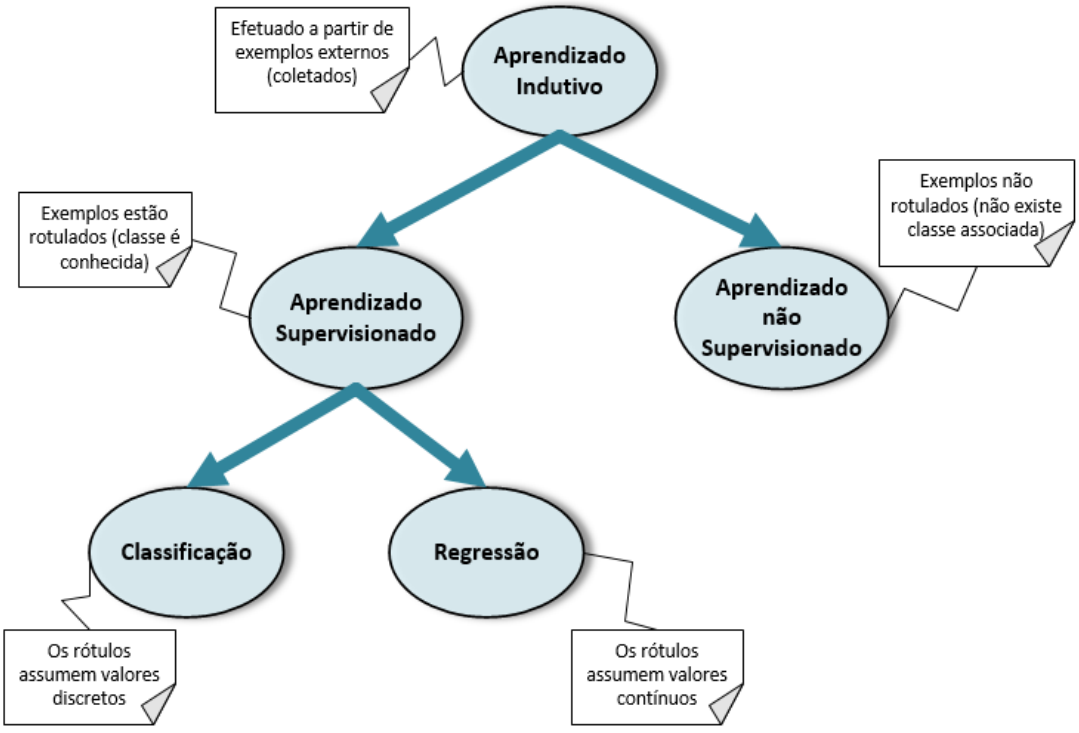


Figura 18 - Aprendizado indutivo
Fonte: Adaptado de BARANAUSKAS (2007)

Segundo KONONENKO e KUKAR (2007) o resultado do aprendizado é conhecimento, e este pode ou não, ser usado pelo sistema para resolver novos problemas. Quando se utiliza sistemas de aprendizado de máquina deve haver a distinção de duas partes: o algoritmo de aprendizado, que por meio de um conjunto de dados de treinamento gera conhecimento (também conhecido como *modelo*) ou modifica o já existente, de modo a se aperfeiçoar; e o algoritmo de execução, que utiliza o conhecimento gerado para encontrar soluções.

O aprendizado é uma das fronteiras atuais dentro da pesquisa de visão computacional e tem recebido maior atenção nos últimos anos. Tecnologias de aprendizado de máquina tem um forte potencial para contribuir para (SEBE *et al.*, 2005):

- Desenvolvimento de algoritmos de visão flexíveis e robustos que irão melhorar o desempenho dos sistemas práticos de visão com um maior nível de competência e maior generalidade
- O desenvolvimento de arquiteturas que vão reduzir o tempo de desenvolvimento de sistemas e propiciar um melhor desempenho.

Para RUSSELL e NORVIG (1995) existem três razões principais para a construção de um algoritmo que aprenda, a primeira delas se refere ao fato de que, quem projeta, não tem como antecipar todas as situações possíveis que um agente (aquele que realiza uma ação) possa encontrar. A segunda diz respeito ao fato de que, os agentes precisam se adaptar as mudanças, já que quem projeta não tem como antecipar essas mudanças. A terceira é que em alguns o projetista não tem uma ideia de solução a ser programada.

Programar um computador para realizar o reconhecimento de faces, por exemplo, só é possível por meio de algoritmos de aprendizagem (RUSSELL e NORVIG, 1995). No problema de categorização, que é um dos focos desse estudo, aprender objetos visuais implica em reunir imagens de treinamento de cada categoria, e, em seguida, extrair ou aprender um modelo que pode fazer previsões para presença ou localização de um objeto em novas imagens. Os modelos são muitas vezes construídos por meio de métodos de aprendizagem de máquina (GRAUMAN e LEIBE, 2011).

Dessa perspectiva o aprendizado de máquina propicia a construção de modelos que possibilitem a análise do que está contido na imagem.

3.2.1 Classificação

Como pode ser visto por meio da Figura 18, a classificação é uma tarefa que pertence ao que é chamado aprendizado supervisionado. De modo geral, a aprendizagem supervisionada é utilizada quando se tem uma boa base de exemplos, cujas instâncias tem suas classes conhecidas.

Segundo BARANAUSKAS e MONARD (2000) o objetivo principal da classificação é criar um classificador que aprenda, que consiga classificar objetos que ainda não foram vistos mas acredita-se que irão pertencer a uma das classes para qual o classificador foi construído.

A aprendizagem é a busca através do espaço de hipóteses possíveis para encontrar aquela que terá um bom desempenho, mesmo com novos exemplos, afora o conjunto de treinamento. Para medir a precisão de uma hipótese é necessário testa-la fornecendo um conjunto de teste de exemplos que são distintos do conjunto de treinamento (RUSSELL e NORVIG, 1995).

O conjunto de dados (Tabela 1) geralmente é representado na forma estrutural de uma matriz, também chamada de tabela atributo-valor, e ele pode ser dividido em duas categorias.

- *Conjunto de Treinamento*: Principal conjunto de informações fornecidas ao indutor para construção do modelo para geração de hipóteses. Na tarefa de Classificação cada exemplo tem uma classe associada.
- *Conjunto de Teste*: Dados que serão utilizados para avaliar o modelo construído a partir do conjunto de treinamento. É recomendável que as instancias deste conjunto não tenham sido utilizadas no treinamento.

Tabela 1 - Tabela atributo-valor

	X_1	X_2	. . .	X_m	Y
T_1	x_{11}	x_{12}	. . .	x_{1m}	y_1
T_2	x_{21}	x_{22}	. . .	x_{2m}	y_2
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
T_n	x_{n1}	x_{n2}	. . .	x_{nm}	y_n

Fonte: BARANAUSKAS e MONARD (2000)

Os conjuntos são construídos a partir de um número de instâncias ($T_1 ... T_n$) também conhecidas como exemplos, formadas por um vetor de atributos ($X_1 ... X_n$) que descrevem algumas características da mesma. Os atributos que são utilizados no trabalho são de dois tipos, os nominais que possuem um valor discreto (Cor: vermelho, verde, azul) e os contínuos, com algum valor real (peso expresso em Kg: 2, 5, 100). Por último, para cada instancia T, existe um rótulo Y associado a ela, este rótulo representa a tarefa que pretende-se aprender, por exemplo “jogar tênis”, “não jogar tênis”. Para a tarefa de *Classificação*, o valor do rótulo sempre será um valor nominal, caso o mesmo seja um valor continuo a tarefa passa a ser denominada *Regressão* (BARANAUSKAS e MONARD, 2000).

É importante na construção do conjunto de treinamento não apenas considerar exemplos positivos. Um algoritmo de aprendizagem precisa conhecer exemplos contrários para ser capaz de distinguir o falso do verdadeiro e predizer corretamente a classe à qual a instância pertence.

Como exemplo, tem-se um conjunto de carros pertencentes a uma “família de carros”, um grupo de pessoas olha para cada carro e os rotula. Os carros que eles acreditam que são carros familiares são exemplos positivos, e os outros carros são exemplos negativos. A aprendizagem se dá justamente em encontrar uma descrição que é compartilhada por todos os exemplos positivos e nenhum dos exemplos negativos. Com isso, é possível fazer a previsão: dado um carro que não foi visto antes, verificando com a descrição aprendida, o algoritmo será capaz de dizer se é um carro da família ou não (ALPAYDIN, 2010).

O processo de classificação é ilustrado na Figura 19, e se dá a partir do pré-processamento de dados brutos para a construção da especificação do problema, os exemplos então são fornecidos para a construção do classificador que é posteriormente avaliado e se for necessário, o processo é repetido de modo a melhorar o desempenho (METZ, 2006).

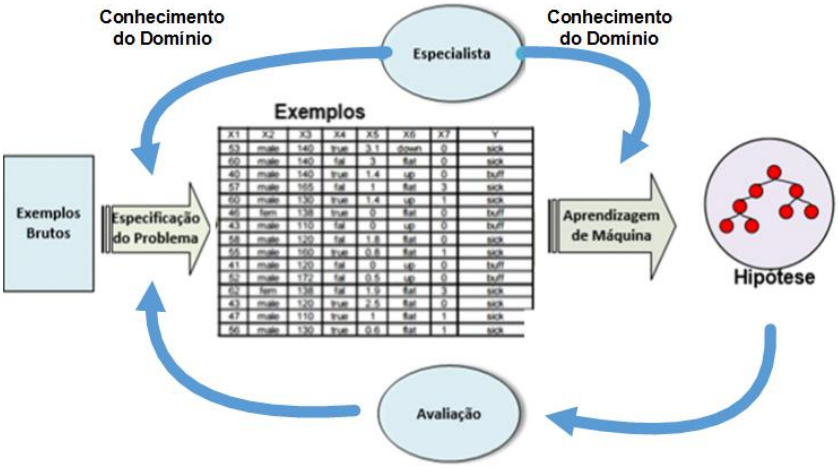


Figura 19 - Classificação
Fonte: BARANAUSKAS (2007)

Treinar o classificador significa determinar o melhor conjunto de atributos para um classificador. Os métodos mais eficazes para treinar classificadores envolvem a aprendizagem a partir de exemplos.

A Figura 20 ilustra um exemplo onde é simulado um caso de classificação de imagens médicas (THEODORIDIS e KOUTROUMBAS, 2006).

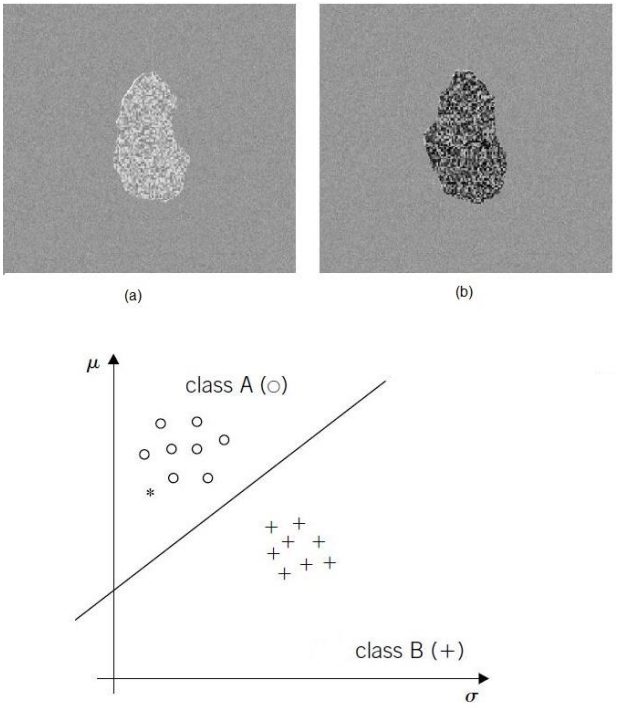


Figura 20 - Exemplo de Classificação
Fonte: Adaptado de THEODORIDIS e KOUTROUMBAS (2006)

Nela é possível ver duas imagens, cada uma com uma região distinta dentro dela. Considerando que a Figura 20a representa o resultado de uma lesão benigna (Classe A) e a Figura 20b representa uma lesão maligna, um câncer (Classe B), e assumindo que as duas imagens não são os únicos padrões disponíveis, mas que existe o acesso a um banco de dados com um certo número de imagens originadas tanto da classe A quanto da B, o primeiro passo é identificar as quantidades mensuráveis que fazem as duas regiões distintas uma da outra (THEODORIDIS e KOUTROUMBAS, 2006).

Cada ponto corresponde a uma diferente imagem do banco de dados. É possível notar que os padrões da classe A tendem a se espalhar em uma área diferente dos padrões da classe B (THEODORIDIS e KOUTROUMBAS, 2006).

Treinar o classificador para realizar esta tarefa, significa fornecer exemplos dos padrões da imagem, para que o mesmo possa identificar com base no que ele conhece novos exemplos de ausência ou presença de câncer. Na imagem, a linha reta que corta o plano pode representar os parâmetros que o classificador descobriu que separam as duas classes e que possibilita-o classificar novas instancias.

3.2.2 Agrupamento de Dados

Em contraste com aprendizado supervisionado, a aprendizagem não-supervisionada ajusta um modelo para observações assumindo que não há variável aleatória dependente, saída ou resposta, ou seja, um conjunto de observações de entrada é recolhida e tratada como um conjunto de variáveis aleatórias e analisadas como tal. Nenhuma das observações é tratada de forma diferente das demais (CLARKE, FOKOUE e ZHANG, 2009).

Ou seja, na tarefa de Agrupamento não existe classe (Y) associada a cada exemplo (ou existe, mas é ignorada). Sendo assim, como alternativa a tabela atributo-valor usada na aprendizagem supervisionada, pode ser utilizada uma outra estrutura para representar o conjunto de dados, baseando-se na similaridade dos exemplos (METZ, 2006).

Tabela 2 - Matriz de Similaridade

Exemplos	E_1	E_2	E_3	...	E_N
E_1	—	$sim(E_1, E_2)$	$sim(E_1, E_3)$	...	$sim(E_1, E_N)$
E_2	$sim(E_2, E_1)$	—	$sim(E_2, E_3)$	...	$sim(E_2, E_N)$
E_3	$sim(E_3, E_1)$	$sim(E_3, E_2)$	—	...	$sim(E_3, E_N)$
⋮	⋮	⋮	⋮	—	⋮
E_N	$sim(E_N, E_1)$	$sim(E_N, E_2)$	$sim(E_N, E_3)$	...	—

Fonte: METZ (2006)

No contexto da falta do rótulo, a aprendizagem não-supervisionada se utiliza de técnicas específicas, baseadas em distância para agrupar as instâncias não rotuladas, de modo que os exemplos mais similares entre si fiquem mais próximos num determinado grupo (*cluster*) e os dissimilares se encontrem em outros grupos.

Considerando que cada exemplo E_N da Tabela 2 é constituído de atributos (i.e. características), a similaridade entre os exemplos é calculada com base nessas informações. Para realizar este cálculo é necessário especificar uma medida de distância. Considerando M, como o número de atributos que descrevem cada exemplo, a seguir são brevemente descritas as medidas de distância mais comuns e utilizadas para o cálculo de similaridade (METZ, 2006):

- *Euclidiana:* Basicamente é a distância geométrica *Euclidiana* entre os exemplos. Sendo um caso especial da distância de *Minkowsky,* onde $r = 2$,

$$dist(E_i, E_{ij}) = \sqrt{\sum_{l=1}^{M} (x_{il} - x_{jl})^2} \quad \textbf{(3)}$$

- *Manhattan:* Também conhecida como *city block*, pois numa cidade é praticamente impossível ir de um ponto a outro em linha reta, seu valor é representado pelo módulo das diferenças dos valores dos atributos (eixos coordenados).

$$dist(E_i, E_{ij}) = \sum_{l=1}^{M} |x_{il} - x_{jl}| \quad \textbf{(4)}$$

- *Minkowsky:* É uma distância considerada uma generalização para cálculo de distância entre dois pontos em espaços *r*-dimensionais. $r = 1$ ela é conhecida como distancia *Manhattan*.

$$dist(E_i, E_{ij}) = \left(\sum_{l=1}^{M} |x_{il} - x_{jl}|^r \right)^{\frac{1}{r}} \quad (5)$$

Apesar de classes e clusters serem diferentes, cada cluster pode comportar um número *x* de exemplos, e estes de tal forma pertencerem a um determinado conceito, que pode ser visto como a classe (METZ, 2006).

Do ponto de vista da aprendizagem de máquina, o processo de construção de classificações é uma forma de "aprender com a observação" (aprender sem um professor). Esta forma de aprendizado de máquina foi estudada sistematicamente em áreas como análise de agrupamentos (*clusters*). A noção central usada lá para a criação de classes de objetos é uma medida numérica de semelhança de objetos. De forma intuitiva os padrões de um *cluster* válido são mais similares entre eles do que os padrões que pertencem a *clusters* diferentes, isto pode ser visto por meio da Figura 21, onde os dados de entrada representados por *x* (Figura 21a) são alocados a diferentes *clusters* (Figura 21b), por meio de alguma técnica que mede a similaridade entre os dados. (JAIN, MURTY e FLYNN, 1999).

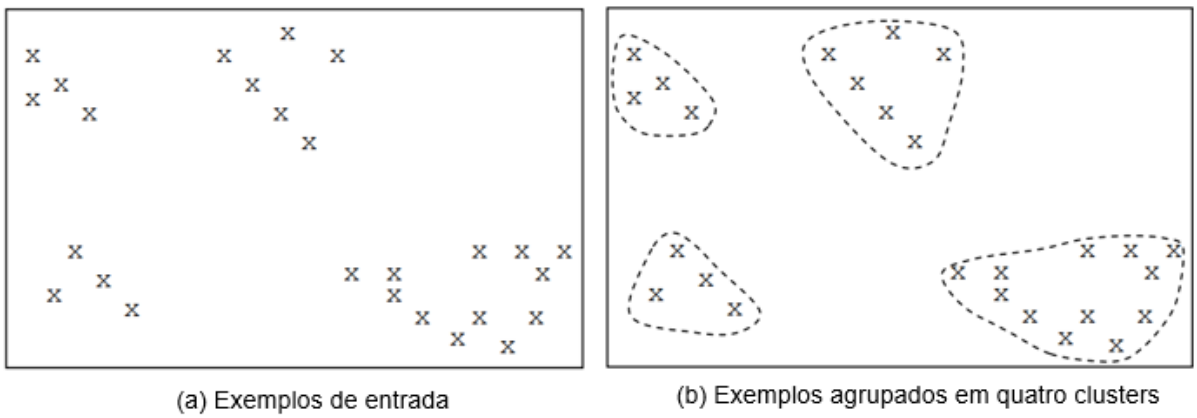


Figura 21 - Exemplo de Agrupamento
Fonte: Adaptado de BRAMER (2007)

3.2.3 Algoritmos

- *Multilayer Perceptron*: Segundo RUSSELL e NORVIG (1995) Rede neural é um modelo matemático que copia a atividade mental do cérebro humano para gerar conhecimento, formada por estruturas chamadas de neurônios artificiais sendo este último formado por ligações, de entrada e saída, uma função de entrada, uma função de saída e uma saída. Esse neurônio é disparado quando uma combinação linear de suas entradas excede algum limiar, ou seja, ele implementa um classificador linear. A organização de uma rede neural é feita em camadas de neurônios interconectados, e essa interconexão é responsável por definir a arquitetura da rede.
- *FURIA*: A lógica fuzzy (ou lógica nebulosa) possui características muito diferentes da lógica tradicional, ao começar pelos possíveis valores que um valor verdade de uma preposição pode assumir. Na lógica fuzzy, esses valores podem ser um subconjunto fuzzy de qualquer conjunto parcialmente ordenado. Além disso, esses valores podem ser expressos linguisticamente, por meio de predicados nebulosos (alto e baixo, por exemplo). São criadas inferências por meio da aplicação de regras *se a então b*. O Classificador por indução de regras nebulosas não ordenadas (FURIA) é um método de classificação baseado em regras, que aprende a separar cada classe de todas as outras classes, o que significa que nenhuma regra padrão é usada e a ordem das classes é irrelevante ou seja quando há o treinamento de um classificador, as outras classes não são consideradas (HÜHN e HÜLLERMEIER, 2009).
- *Random Forest*: São uma combinação de árvores de predição, de tal modo que cada árvore depende dos valores da amostra de um vetor randômico independente e com a mesma distribuição para todas as árvores na floresta. De modo geral é feita a combinação de árvores. Ela funciona com a construção de cada árvore se baseando em uma amostra do conjunto de treinamento, a partir da construção das mesmas, é realizado uma votação com a escolha de cada árvore, portanto, a classe final é escolhida com base no maior número de votos. Mais detalhes podem ser encontrados em (BREIMAN, 2001).
- *kNN (k-nearest neghbor)*: é um algoritmo de classificação baseado na distância dos exemplos em relação à similaridade de suas características. O algoritmo

procura por *k* instâncias no conjunto de treinamento que sejam mais similares com o objeto de teste, funcionando da seguinte maneira (WU *et al.*, 2008):

i) É calculado a distância entre o exemplo de teste e os exemplos do conjunto de treinamento;
ii) Identifica-se os *k-vizinhos* mais próximos;
iii) A classe majoritária entre os *k-vizinhos* é associada ao exemplo de teste.

Portanto, dois fatores importantes na configuração deste algoritmo são o número de vizinhos (*k*) e a medida de distância.

- *Support Vector Machines (SVM)*: Traduzidas como máquinas de vetores de suporte, possuem fundamento baseado em teorias estatísticas e tem como principal objetivo achar a melhor função de classificação para distinguir exemplos de duas classes. Isto é feito por meio da construção de um hiperplano ótimo para separar as classes. Mais informações a respeito das SVM's podem ser encontradas em (LORENA e DE CARVALHO, 2003).
- *K-Means:* Funciona de forma iterativa para particionar um conjunto de dados e um número específico de grupos (*clusters*). Ele funciona da seguinte maneira (WU *et al.*, 2008):
    i) É definido um número especifico *n* de *clusters* (grupos).
    ii) *k* objetos denominados centroides dos *clusters* são criados arbitrariamente (forma aleatória).
    iii) Para cada elemento do conjunto de dados, atribui-se ao mesmo o centroide mais próximo, baseando-se na distância.
    iv) Após cada elemento estar associado à um dos *k* centroides, significa que estarão formados *n* grupos.
    v) Recalcula-se o valor dos *k* centroides como sendo a média de todos os elementos associados a ele (elementos do grupo).
    vi) Repete-se o processo até que os centroides não sejam mais modificados.
    vii) Tem-se ao final *n* grupos formados.

3.3 BAG-OF-KEYPOINTS

O *Bag-of-Keypoints*, também conhecido como *Bag-of-Features* é um modelo que utiliza princípios de outro modelo chamado *Bag-of-Words,* muito aplicado na área de *Recuperação de Informação* (SALTON e MCGILL, 1986). O termo do modelo foi proposto primeiramente por CSURKA *et al.* (2004) e utilizado para a tarefa de categorização visual, porém, SIVIC e ZISSERMAN (2003) já haviam utilizado anteriormente o modelo *Bag-of-Words,* na criação de um vocabulário visual por meio de descritores para recuperação de vídeo.

A metodologia *Bag-of-words* foi proposta pela primeira vez para a análise de documentos de texto e posteriormente adaptado para aplicações de visão computacional. Os modelos são aplicados a imagens usando uma visualização análoga a uma palavra, formado pelo vetor que quantiza as características visuais (cor, textura, etc.) como descritores de região (BOSCH, MUÑOZ e MARTÍ, 2007).

A principal diferença desse modelo utilizado no reconhecimento genérico para o reconhecimento de instancia, é a ausência de uma verificação geométrica uma vez que as instancias individuais das categorias, possuem pouca coerência espacial em relação as suas características (SZELISKI, 2010), e isto é uma das principais desvantagens.

Segundo CSURKA *et al.* (2004) é necessário entender que há diferença entre categorização visual e detecção (proposta deste trabalho), e que apesar desta abordagem ser um tanto trabalhosa para a tarefa de detecção, devido a necessidade de construção de uma modelo para cada categoria, é possível aplicá-la.

3.3.1 Construção do Modelo

3.3.1.1 Treinamento

O primeiro passo do processo para criação do modelo é obter um conjunto de imagens exemplos, que deverão estar previamente categorizadas, para que ao final do processo da construção da nova representação da imagem, a categoria esteja indexada a imagem. O processo para a construção do vocabulário e treinamento é resumido no pseudo-algoritmo a seguir (CSURKA *et al.*, 2004):

- Detecção e descrição dos pontos chaves de cada imagem.
- Construção do Vocabulário
    - Agrupar todos pontos chaves em k clusters
- Para cada imagem computar o histograma de frequências com base no vocabulário construído.
- Treinar um classificador tendo como entrada a nova representação de cada imagem (um vetor construído a partir do histograma de frequências).

É realizado a detecção das regiões de interesse, sejam elas, texturas, cor ou como foi visto no Capitulo 2, regiões de máximos e mínimos baseadas em Diferenças da aplicação do filtro Gaussiano. Contudo, para descrição, é preferível utilizar descritores locais, devido as propriedades de invariância a perspectiva, escala, mudança na iluminação e oclusões, para que seja possível carregar o máximo de informação possível para discriminação no processo categorização (CSURKA *et al.*, 2004).

Com todos os descritores de pontos chaves (i. e. vetor de característica extraído representando cada região local), extraídos de cada imagem do conjunto de treinamento, é criado o vocabulário de palavras visuais, que é um único vetor que representará a imagem. O vocabulário é construído por meio de um algoritmo que realiza o Agrupamento de todos os descritores extraídos, por exemplo, por meio do algoritmo *K-Means*, cada informação visual da imagem é representado por cada centroide do *cluster* (CSURKA *et al.*, 2004).

Em seguida os descritores de cada imagem são novamente *clusterizados* utilizando como base os centroides do *cluster* gerado, ao final é contado o número de

ocorrências de pontos em cada centroide para a construção de um histograma de frequências.

O histograma de frequência pode ser representado como um vetor, concatenando todos os valores, onde posteriormente é fornecido para um algoritmo de aprendizado supervisionado, servindo como exemplo para construção de um modelo de classificação. Para cada categoria de objeto que se deseja reconhecer é necessário a construção de um modelo.

O processo é apresentado por meio da Figura 22.

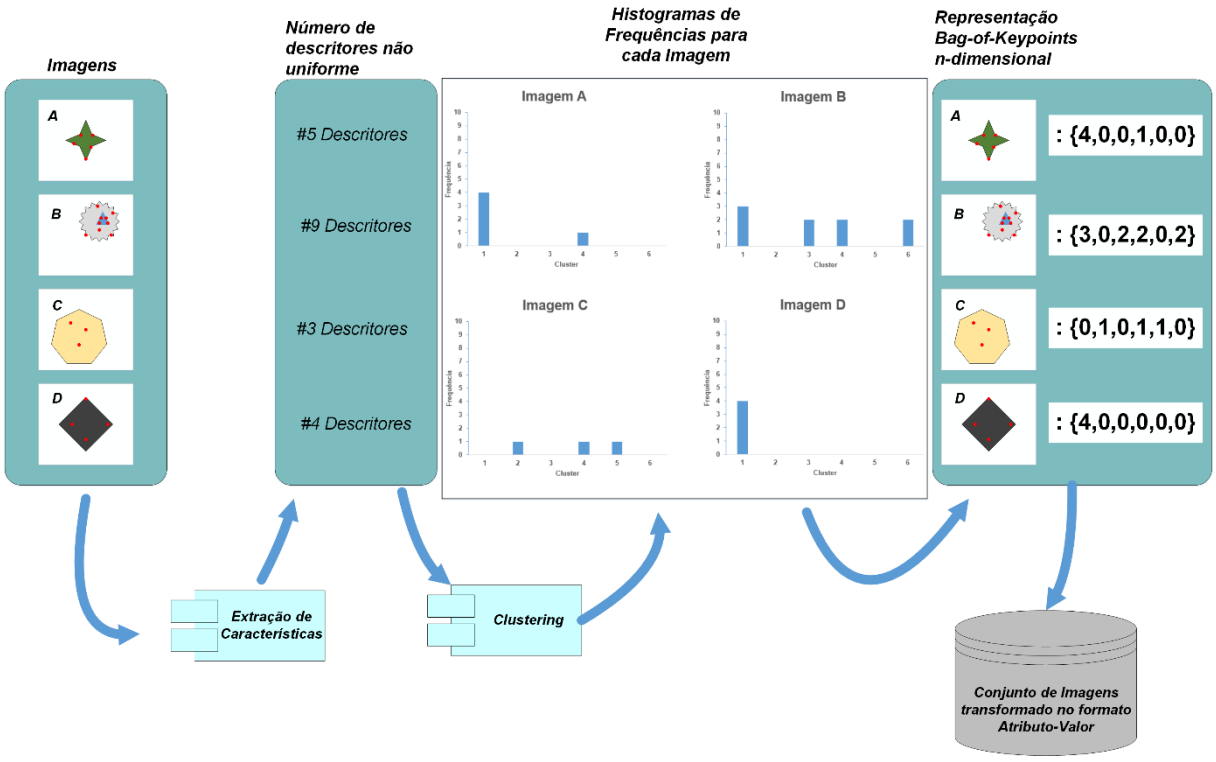


Figura 22 - Ilustração do processo de obtenção do Bag-of-Words
Fonte: GHELLERE et al. (2015).

Por exemplo, supondo que 5 características tenham sido detectadas e descritas de uma imagem e que o tamanho do nosso vocabulário visual (*cluster*) seja 3. Após o agrupamento a distribuição dessas características em cada grupo (*cluster*) tenha ficado da seguinte maneira:

- *Cluster* 1: 3 características.
- *Cluster* 2: 0 características.
- *Cluster* 3: 2 características.

A nova representação da imagem será a concatenação das frequências de suas características em cada cluster, ou seja, {3,0,2}.

3.3.1.2 Detecção via Classificação

A partir do momento que se tem o vocabulário de palavras visuais construído, para cada nova imagem o qual se deseja detectar um objeto, extraem-se os pontos chaves e a partir do *cluster* gerado monta-se o histograma de frequência com as ocorrências dos pontos em cada centroide.

O vetor formado pela concatenação dos valores do histograma é fornecido então, ao classificador modelado que irá categorizar a imagem, baseando-se nos parâmetros do vetor.

3.3.2 Considerações

Existem algumas considerações a serem feitas. Os passos descritos anteriormente são os principais para se construir uma nova representação das imagens por meio do uso do modelo *Bag-of-Keypoints* que serão fornecidas aos indutores para criação de um modelo de classificação.

Mas para a construção da representação existem escolhas que impactam consideravelmente. A primeira delas, diz respeito a qual método de detecção e descrição de características utilizar, isto é importante, pois os pontos extraídos devem possuir características invariantes, no decorrer deste trabalho foram citados e explicados os algoritmos SIFT e SURF.

Em seguida é necessário decidir qual método para quantização será utilizado, foi citado como exemplo, o uso do algoritmo de Agrupamento *k-Means*, mas o simples fato de escolhe-lo acarreta na definição de outros parâmetros do próprio algoritmo que poderão influenciar de forma considerável o modelo, como por exemplo, o número de cluster a ser utilizado ou mesmo a função de distância.

Por último, qual algoritmo de classificação utilizar. Já foi apontando que não existe um modelo que seja eficiente para todo e qualquer domínio de problema. Encontrar o melhor modelo que se ajuste e traga mais eficiência a tarefa, é uma das etapas, e isto é feito mediante experimentos, testes e avaliação.

3.4 AVALIAÇÃO

A avaliação do modelo é feita avaliando-se os resultados finais do modelo de classificação construído. A avaliação é muito importante, tanto para realizar comparações entre algoritmos de classificação, verificando o desempenho de cada um em relação a um determinado domínio de problema como para realizar a verificação da confiança da taxa de acerto e erro do classificador quando o mesmo for aplicado numa situação real (ALPAYDIN, 2010; LAVESSON, 2006).

As informações de avaliação são obtidas por meio do uso de um conjunto de instâncias denominado *Conjunto de Teste*, como já foi mencionando anteriormente no trabalho. Nas sessões a seguir serão abordadas brevemente algumas medidas para verificar o desempenho dos classificadores assim como modelos para realizar a avaliação.

3.4.1 Matriz de Confusão e Medidas de Avaliação

Para um classificador sobre um conjunto de exemplos T, sua matriz de confusão traz informações referentes ao número de classificações preditas corretamente em relação a classe referência. O número de acertos, para cada classe, se localiza na diagonal principal da matriz $M(Ci, Cj)$ para $i \neq j$, representam erros na classificação.

Um classificador ideal, portanto, possui uma matriz de confusão onde todos os elementos nas células da matriz (Ci, Cj) com $i \neq j$ iguais a zero, o que significa que ele não comete erros.

A matriz de confusão é apresentada por meio da Tabela 3.

Tabela 3 - Matriz de confusão

	Classe Prevista		
Classe Verdadeira	Previsto (Positivo)	Previsto (Negativo)	Total
Real (Positivo)	tp	fn	p
Real (Negativo)	fp	tn	n
Total	p'	n'	T

Fonte: Adaptado de ALPAYDIN (2010)

A partir da Tabela 3, específica para problemas de duas classes, pode-se obter algumas informações. Para um exemplo positivo, se o valor predito também for positivo, isso significa que o mesmo é um *true positive* (tp), se a predição é negativa para um exemplo positivo, tem-se um *false negative* (fn). Para um exemplo negativo, se a predição também for negative, significa que tem-se um *true negative* (tn), e tem-se um *false positive* (fp) se o valor predito para um exemplo negativo for positivo.

Com estas informações é possível realizar o cálculo de algumas medidas para avaliação da performance de um classificador, no caso do presente trabalho, baseado em problema de duas classes. As medidas são descritas brevemente a seguir (ALPAYDIN, 2010):

- *Erro (Error):* Medida que representa o percentual de erros que o classificador comete.

$$error = \frac{fp + fn}{T} \quad (6)$$

- *Acurácia (Accuracy)*: Percentual de acertos do classificador sobre todos os exemplos do conjunto.

$$accuracy = \frac{tp + tn}{T} = 1 - error \quad (7)$$

- *Taxa de Verdadeiros Positivos (tp-rate)*: taxa de classes válidas corretamente classificadas.

$$tp_{rate} = \frac{tp}{p} \quad (8)$$

- *Taxa de falsos alarmes (fp-rate)*: é a taxa de classes ‘impostoras’ que são incorretamente aceitas.

$$fp_{rate} = \frac{fp}{n} \quad (9)$$

- *Precisão (Precision)*: esta medida determina a eficácia do classificador em reconhecer as instancias de uma classe de interesse e descartar as demais. É o percentual de exemplos classificados corretamente como positivos dentre todos os que foram classificados como positivos.

$$precision = \frac{tp}{tp + fp} \quad \textbf{(10)}$$

- *Sensitividade (Recall)*: esta medida determina a eficácia do classificador em reconhecer todas as instancias de uma classe de interesse. É o percentual de exemplos classificados como positivos dentre todos os que são realmente positivos.

$$recall = \frac{tp}{tp + fn} \quad \textbf{(11)}$$

- *Matthews correlation coeficiente (MCC):* Também conhecido como coeficiente *Phi* essa medida avalia para problemas binários, a qualidade da classificação. Seu valor varia de -1 à +1, sendo que +1 representa a perfeita predição, 0 representa que o classificador não é melhor que uma predição aleatória e -1 representa o desacordo entre predição e observação (PARKER, 2011):

$$MCC = \frac{tp * tn - fp * fn}{\sqrt{(tp + fp) * (tp + fn) * (tn + fp) * (tn + fn)}} \quad \textbf{(12)}$$

3.4.2 Curvas ROC (Receiver Operating Characteristic)

As curvas ROC permitem realizar a comparação, organização e até mesmo selecionar classificadores diferentes por meio de uma visualização gráfica de seu desempenho, o que torna mais inteligível interpretar e compreender a dimensionalidade do problema de avaliação (FAWCETT, 2004). Esse modelo de avaliação se baseia na probabilidade de detecção, ou seja, taxa de verdadeiros positivos (tp_{rate}) e na probabilidade de falsos alarmes, taxa de falsos positivos (fp_{rate}), plotando se o primeiro no eixo das abcissas *y* e o segundo no eixo das ordenadas *x.*

Dado um classificador discreto aquele cuja saída é apenas uma classe, ele produz um par (fp_{rate}, tp_{rate}), que corresponde a único ponto no espaço ROC, representado na Figura 23 (FAWCETT, 2004).

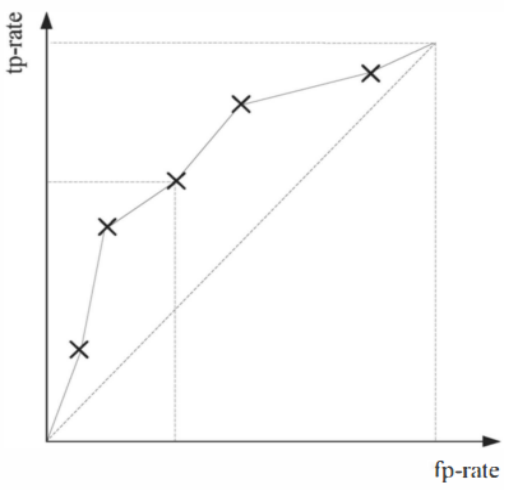


Figura 23 - Exemplo do espaço ROC.
Fonte: ALPAYDIN (2010)

Na Figura 23 é possível visualizar um exemplo da curva ROC e seu espaço, tanto o eixo *x* quanto o eixo *y* variam no intervalo $[0,1]$. Pontos que pertencerem ao triangulo superior à linha diagonal que parte da origem em direção ascendente, representam classificadores que se saem melhor do que o acaso, em contra partida, modelos pertencentes ao triângulo inferior representam aqueles que saem piores que a aleatoriedade (PRATI, BATISTA e MONARD, 2008).

Uma outra importante forma de mensurar a performance de um classificador é reduzir a curva ROC a único valor. A área abaixo da curva ROC (AUC) é uma parte da área do espaço ROC, ou seja, ela sempre estará entre 0 e 1, onde o classificador ideal possui uma $AUC = 1$. Na Figura 24 é possível observar as duas diferentes áreas de diferentes modelos.

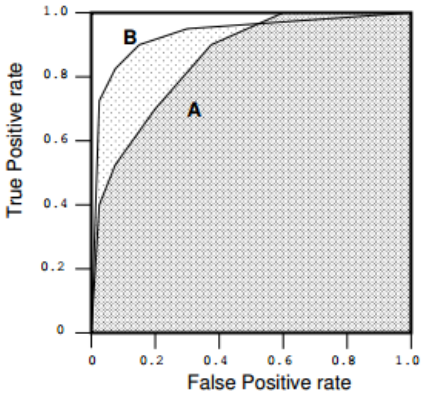


Figura 24 - Exemplo gráfico da medida AUC.
Fonte: FAWCETT (2004)

3.4.3 Validação

Esta técnica busca avaliar a eficácia da generalização do modelo de classificação para predizer para novos exemplos, ou seja, instancias que não fizeram parte do treinamento. Para isso é realizado o particionamento dos dados. É obtido um conjunto de pares de treinamento e validação de um conjunto de dados T, dividindo-se randomicamente o conjunto em *K* partes. De modo randômico também é dividido cada *K*-parte em duas, usando metade para treinamento e a outra metade para validação (ALPAYDIN, 2010). Está é apenas uma, das muitas configurações que podem ser utilizadas para validação cruzada, a seguir serão descritos brevemente as principais formas de particionamento e iteração desta técnica (ALPAYDIN, 2010; KOHAVI, 1995):

- *Holdout*: também conhecido como teste de amostragem, particiona os dados em duas sub-partes mutualmente exclusivas, chamadas de conjunto de treinamento e conjunto de teste. O mais usual é realizar a divisão deixando $^2/_3$ do número de exemplos do conjunto total (T) para o conjunto de treinamento e $^1/_3$ para o conjunto de teste. Para fugir do cenário em que a acurácia possa ser influenciada por um possível padrão na partição que constitui o conjunto de teste, é possível aplicar o *holdout* diversas vezes (*n* vezes), de modo que a taxa de acertos é obtida por meio da média entre *n* testes realizados (KOHAVI, 1995). Na Figura 25 é possível observação a exemplificação do método.

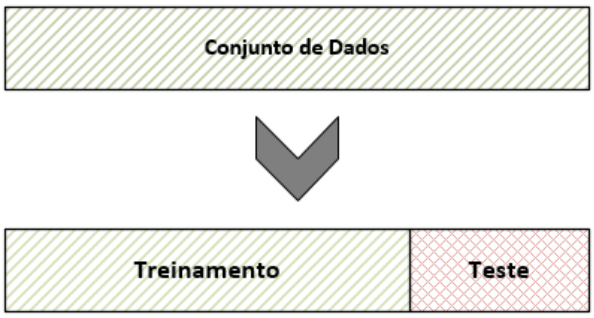


Figura 25 - Exemplificação do Método *Holdout*
Fonte: Adaptado de ALPAYDIN (2010)

- *K-fold Cross-Validation*: nesta abordagem o conjunto de dados T é randomicamente dividido em *k* subconjuntos mutualmente exclusivos, com tamanhos aproximadamente iguais. O classificador é treinado e testado *k* vezes, sendo que *k-1* subconjuntos são combinados e utilizados para treinamento e o restante para teste. O valor de *k* é tipicamente 10 ou 30. Conforme o valor do mesmo cresce, o percentual de instancias de treinamento aumenta conseguindo um classificador mais robusto, porém, o conjunto de validação se torna menor. Também há o custo de construir o modelo de classificação *k* vezes (ALPAYDIN, 2010). O método de k-validação cruzada é ilustrado por meio da Figura 26.

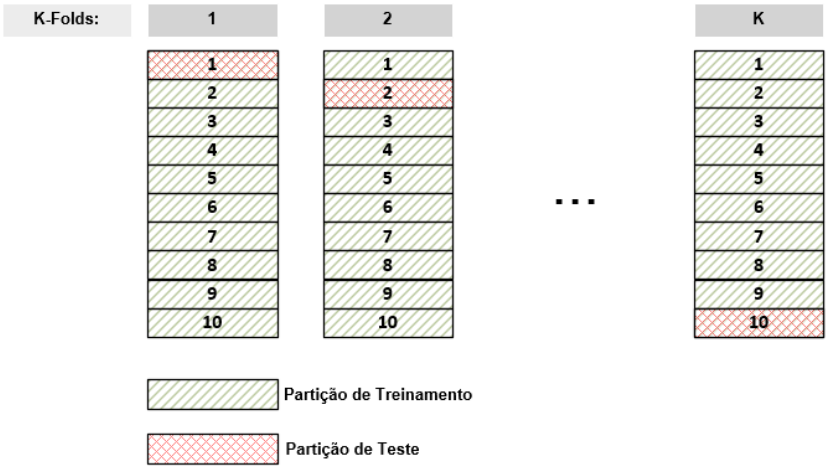


Figura 26 - Exemplificação do método *cross-validation*
Fonte: Adaptado de ALPAYDIN (2010)

- *Leave-one-out*: É o caso extremo do método *k-fold cross-validation* onde o conjunto de dados T (contendo *n* instâncias), é dividido em *n* partes, sendo que

uma parte é deixada de fora para teste e o restante é utilizada para treinamento (ALPAYDIN, 2010). O método é ilustrado na Figura 27.

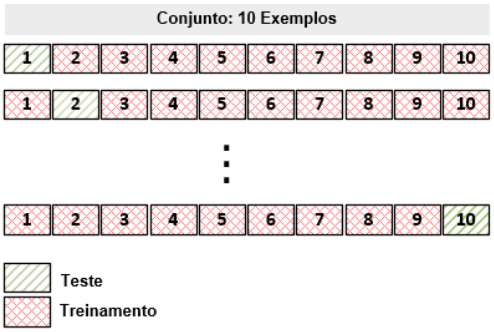


Figura 27 - Exemplificação do método *leave-one-out*
Fonte: Adaptado de ALPAYDIN (2010)

4 MATERIAL E MÉTODOS

Este capítulo apresenta a base para o entendimento acerca da construção do módulo para a classificação de imagens, fornecendo primeiramente uma visualização geral do projeto e em seguida uma breve explicação das bases de dados e ferramentas utilizadas em seu desenvolvimento, o qual foi realizado com base em Linguagem de programação JAVA versão SE 8, levando em consideração a utilização dos conceitos do Paradigma de Orientação a Objetos.

4.1 VISÃO GERAL DA DETECÇÃO VIA CLASSIFICAÇÃO

Os algoritmos de aprendizagem necessitam de um conjunto de dados mapeados na forma de atributos-valor (onde a classe pode ou não estar especificada dependendo da tarefa), onde os atributos devem estar previamente definidos.

Reiterando e considerando que o número de características que são detectadas em cada imagem e posteriormente descritas não é uniforme, houve a necessidade de se escolher um modelo para converter as imagens numa representação que fosse compatível com os padrões da API weka. Por esse motivo, após estudo, o modelo *Bag-of-Keypoints* descrito na Sessão 3.3 foi escolhido para criação dessa representação, devido ao mesmo ser bastante conhecido e aplicado no meio cientifico, além de ser relativamente simples.

Na Figura 28 é possível observar uma visão geral do processo para construir um modelo para classificar uma imagem como contendo ou não determinado objeto.

O processo foi dividido em duas partes: Construção do Modelo e Detecção via classificação. No primeiro passo do processo de construção do modelo, cada imagem contida num diretório arbitrário é carregada, sendo a base de dados que servirá como conhecimento de referência para o indutor, ou seja, é o conjunto de treinamento. As imagens segundo a implementação, devem estar previamente nomeadas segundo a presença ou ausência de determinado objeto, por exemplo, *temFlor (001)* e *ausenciaFlor (001)* para que em seguida sobre cada uma, seja aplicado um algoritmo para a

detecção de características locais e um algoritmo para descrição das características, ou seja um descritor.

Tendo como entrada todos os descritores das imagens e o número de clusters, o componente *Bag-of-Keypoints,* então, realiza a construção das novas representações de cada imagem, salvando o *cluster*, que é necessário para a segunda parte do processo. Ao final da primeira parte do processo obtém-se um conjunto de dados transformados, onde as imagens foram transformadas e agora são representadas por um conjunto de atributos (frequências do histograma) associados à sua classe (*temFlor, ausenciaFlor).*

Este novo conjunto de dados no formato atributo-valor é fornecido à um algoritmo indutor para realizar a construção de um modelo de classificação. Com o modelo construído, para cada imagem que chega para ser analisada, ela deve passar pela segunda parte do processo.

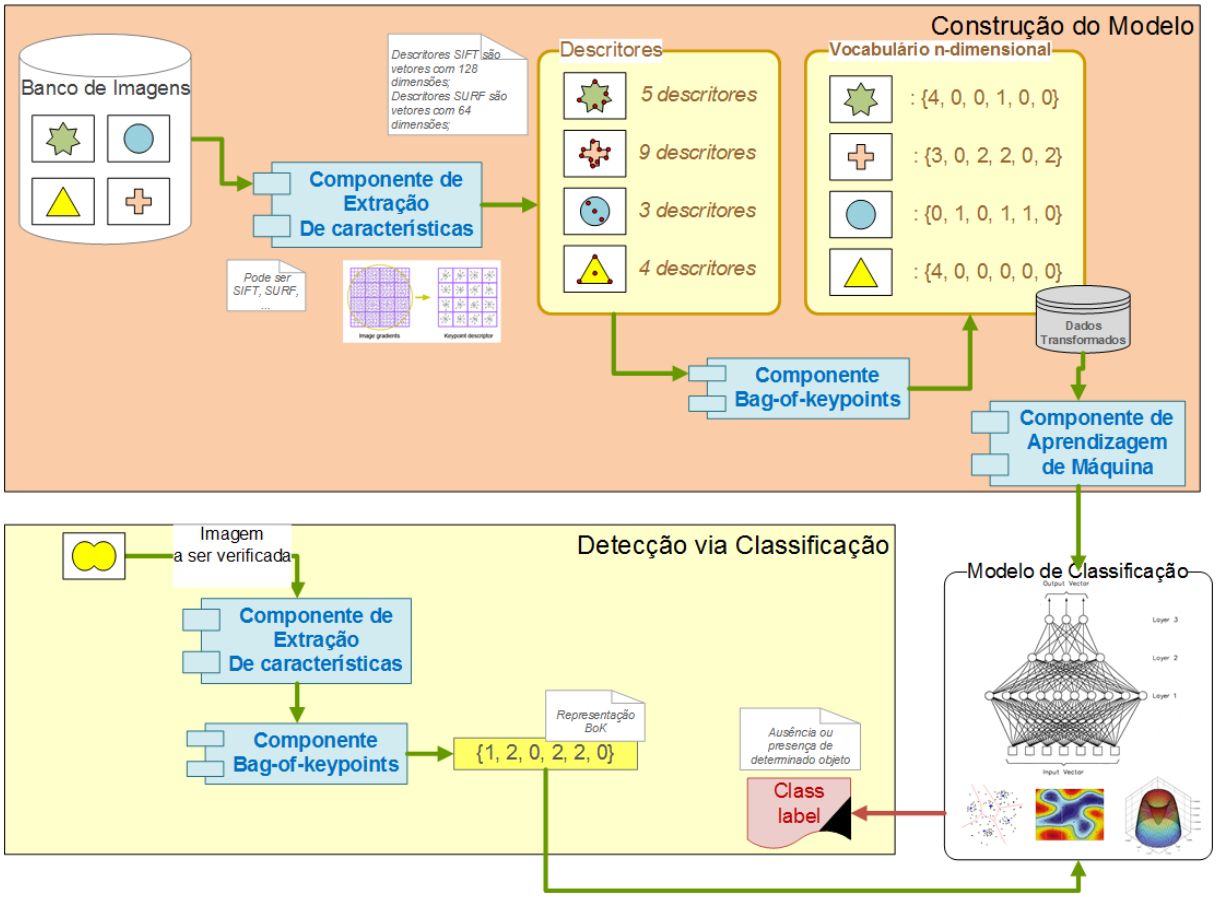


Figura 28 - Visão geral do processo.
Fonte: Autoria Própria.

A imagem a ser analisada, passa pelo componente de detecção e descrição das características. Em seguida com o auxílio do *cluster* salvo na primeira parte, é construída a nova representação da imagem, que nada mais é, do que as frequências concatenadas do histograma obtido com base na alocação das características da imagem que foram alocadas a cada *cluster*. Essa nova representação então é fornecida ao modelo de classificação construído, e o mesmo retorna a classe predita para a imagem (*temFlor, ausenciaFlor).*

4.2 BASE DE DADOS

Para este trabalho foram consideradas 3 bases de dados para construção de alguns modelos para detecção de objetos, sendo escolhidas devido ao número de experimentos, sendo que dois subconjuntos de duas bases distintas foram utilizadas como exemplos positivos e um terceiro subconjunto de uma terceira base de dados foi utilizado como exemplos negativos.

O primeiro conjunto é chamado *Caltech 101*[3] que contém 101 classes de objetos (bola, saxofone, cadeira, câmera, avião etc), tendo uma certa quantidade de exemplos para cada uma das classes, que varia entre 40 e 800, as dimensões aproximadas de cada imagem são de 300x200 *pixels*.

Foram consideradas apenas as imagens presentes na categoria *saxophone* para construção de um modelo para detecção de saxofones. Os exemplos das imagens desse subconjunto podem ser observados na Figura 29.

O *dataset* contém uma outra classe o qual serve para testes e validações, cerca de 480 imagens aleatórias retiradas da web, que foi utilizado para compor exemplos negativos. Mais informações a respeito do conjunto de dados podem ser encontradas em (FEI-FEI, FERGUS e PERONA, 2004).

[3] Disponível em http://www.vision.caltech.edu/Image_Datasets/Caltech101/

Figura 29 - Exemplos de imagens contendo saxofone.
Fonte: Adaptado de FEI-FEI, FERGUS e PERONA (2004).

O segundo conjunto utilizado foi o *GRAZ-02 database*[4] (Figura 30) que possui 3 categorias de objetos, sendo 365 imagens com bicicletas, 311 com pessoas, 420 imagens contendo carros e 380 imagens não contendo as categorias citadas (OPELT *et al.*, 2006). Foram consideradas apenas as imagens com presença de carros para construção de um modelo para detecção de carros.

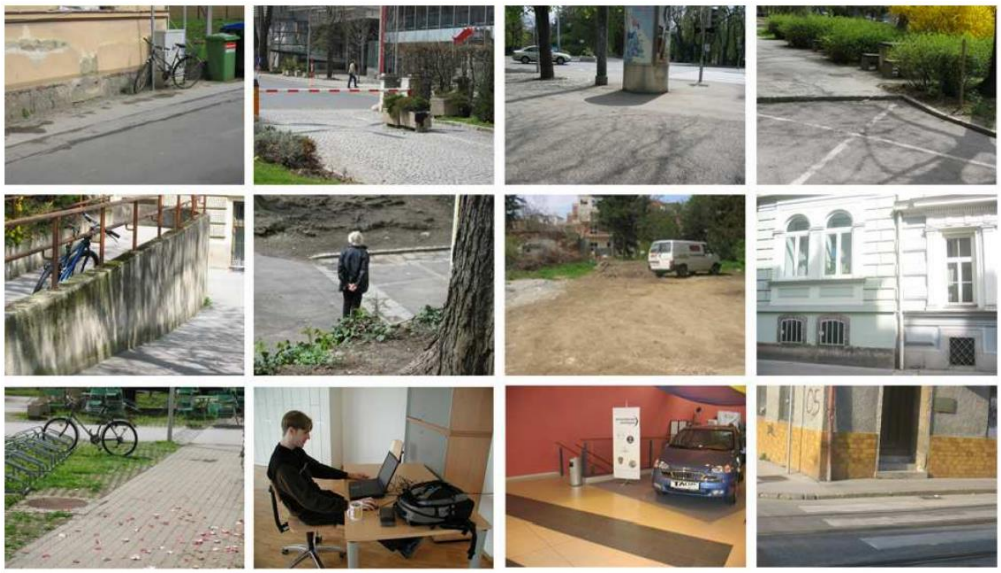


Figura 30 - Exemplos de imagens do *GRAZ-02 database*
Fonte: OPELT *et al.* (2006)

[4] Disponível em http://www.emt.tugraz.at/~pinz/data/GRAZ_02/

O outro conjunto de dados foi utilizado para representar os exemplos negativos na construção de cada um dos modelos, o FlickrLogos-32[5], que é composto por 8240 imagens contendo 32 classes de logos retiradas do Flickr[6], sendo que deste total, 6mil instâncias são de imagens que não contém nenhum logo, e que servem como conjunto de negativos para esse conjunto de dados, sendo assim, este subconjunto foi utilizado para servir de entrada representando exemplos negativos para construção dos modelos de detecção, tendo o cuidado de remover imagens que continham a presença de objetos da classe ao qual estaria servindo como contra exemplo, ou seja, não ter a presença do objeto. Mais informações a respeito do FlickrLogos-32 *dataset* pode ser encontrado em (ROMBERG *et al.*, 2011).

4.2.1 Módulo para download de imagens do Instagram

Para realizar testes em um cenário real, foi utilizado um módulo[7] desenvolvido pelo projeto SNORKEL (*Social network object recognition, knowledge extraction and Learning*)[8], para realizar download de imagens provenientes da rede social Instagram[9].

Para adquirir as imagens, é repassado uma *tag* e o módulo de download devolve as imagens mais recentes contendo a *tag*.

4.3 DETECÇÃO E DESCRIÇÃO DAS CARACTERÍSTICAS

Para a tarefa de detecção características e descrição foi escolhido a biblioteca de visão computacional OpenCV (*Open Source Computer Vision Library*), originalmente desenvolvida pela Intel. Ela possui código fonte aberto para uso acadêmico e comercial, e foi escrita em C e C++ contando com mais de 500 funções que podem

[5] Disponível em http://www.multimedia-computing.de/flickrlogos/
[6] Site: https://www.flickr.com/
[7] Disponível em https://github.com/SnorkelApp/image-downloader
[8] Disponível em https://github.com/SnorkelApp
[9] Site: https://instagram.com/

ser utilizadas em aplicações de diversas áreas tais como: medicina, segurança e inspeção de produtos em fabricas (BRADSKI e KAEHLER, 2008).

Foi escolhida para realização do projeto a biblioteca nativa do OpenCV para Java que acompanha os pacotes disponibilizados. A versão utilizada foi a 2.4.9, nela estão presentes todos os algoritmos responsáveis por extração e descrição de características que foram utilizados neste trabalho.

Para fazer uso dos algoritmos por meio do ambiente de desenvolvimento *Eclipse* versão *Luna* foi criada uma biblioteca de usuário, o qual na sequência foi adicionada ao projeto.

No modulo implementado cada imagem passa por um pré-processamento na qual o canal de cor é convertido para 8-bits (cinza) antes de ter as características detectadas. Tal conversão é realizada com funções presentes no OpenCV. Na Figura 31 é possível observar exemplos[10] da detecção de pontos feito pelos dois algoritmos de detecção de características locais, descritos no Capítulo 2, SIFT e SURF, onde cada círculo vermelho representa um ponto chave detectado.

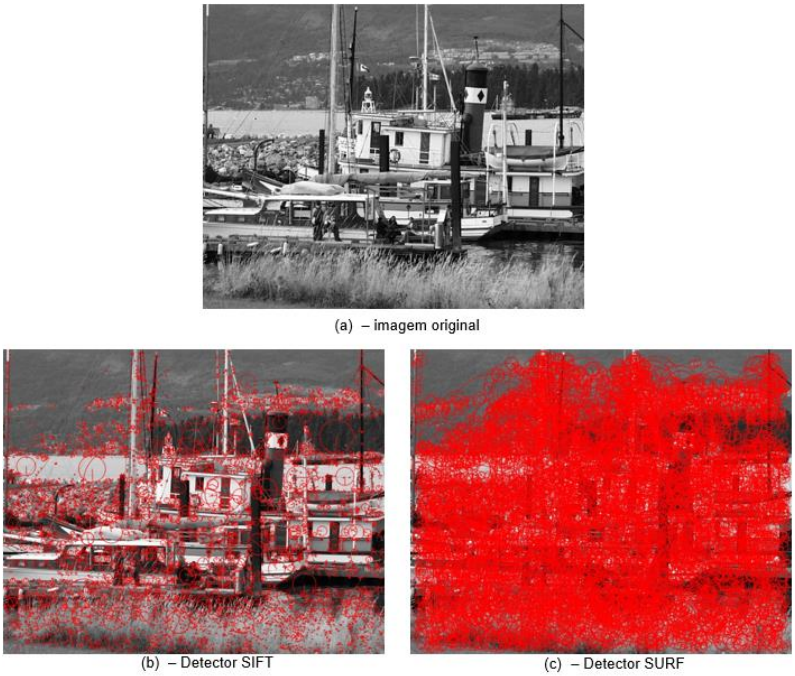


Figura 31 - Exemplos de detecção de pontos chaves
Fonte: Autoria Própria.

[10] Disponível em http://www.robots.ox.ac.uk/~vgg/data/data-aff.html

Para realizar a detecção, utiliza-se a classe `FeatureDetector`[11] instanciando o objeto responsável pela detecção por meio do método estático `create`, passando como parâmetro qual detector se deseja criar (SIFT ou SURF). Em seguida é invocado o método `detect` do objeto instanciado, passando como parâmetros a imagem e um `MatOfKeyPoint`[12], que é um objeto próprio do OpenCV onde ficarão armazenados os pontos detectados.

Para realizar a descrição dos pontos-chaves detectados, é utilizado a classe `DescriptorExtractor` presente no mesmo pacote dos detectores e construído da mesma forma. Para computar os descritores é invocado o método `compute` do objeto passando como parâmetros a imagem, os pontos-chaves detectados e uma Matriz própria do OpenCV, um objeto `Mat`[13] onde ficarão armazenados os vetores que descrevem os pontos chave, ou seja, cada linha da matriz representa um descritor (vetor de características).

Todos os algoritmos utilizados no trabalho tiveram seus parâmetros originais preservados, ou seja, nenhuma modificação, ajuste ou otimização foi feito sobre qualquer um deles.

É importante salientar que em cada imagem são detectados e descritos um número variável de regiões locais (pontos-chaves), isto porque, o número de descritores extraídos depende muito da quantidade de informação contida na imagem.

4.4 ALGORITMOS INDUTORES

Para as tarefas de agrupamento e classificação foi escolhida a *Waikato Environment for Knowledge Analysis* (WEKA) em sua versão de desenvolvedor 3.7.x, utilizando como uma biblioteca para a linguagem Java. A WEKA provém um conjunto de algoritmos de aprendizagem de máquina e ferramentas para pré-processamento de dados, que são muito utilizados em tarefas de *Data Mining*.

[11] Disponível no pacote `org.opencv.features2d`
[12] Disponível no pacote `org.opencv.core.MatOfKeyPoint`
[13] Disponível no pacote `org.opencv.core.Mat`

O projeto WEKA foi fundado pelo governo da Nova Zelândia em 1993, onde no começo o foco de desenvolvimento se dava em linguagem C com rotinas de validação escritas em Prolog (HALL *et al.*, 2009).

Atualmente ele é desenvolvido em Java, e está sob a licença GPL (General Public License).

O motivo que levou a escolha desta biblioteca, além do fato dela possuir o código aberto, foi o grande número de algoritmos implementados e de se ter a facilidade de trabalhar com a linguagem Java. O Weka possui uma comunidade de usuários bastante forte, com muitas aplicações no meio cientifico, contando também com uma documentação muito bem elaborada o que facilita para consultas a respeito de seu funcionamento.

Para a utilização dos algoritmos de aprendizado, os mesmos necessitam de um conjunto dados como referência. A Weka trabalha com diversos formatos dos conjuntos de dados, dentre eles CVS, LibSVM e C4.5 e o formato próprio da API, o *Attribute Relation File Format* (ARFF).

Foi escolhido, portanto, o padrão ARFF[14] (Figura 32) como formato final do conjunto de dados, a ser fornecido como referência para os indutores. Isto por facilitar a manipulação e criação de dados em código trabalhando com objetos próprios da API.

```
1  @relation galaxias
2
3  @attribute 0 numeric
4  @attribute 1 numeric
5  @attribute 2 numeric
6  @attribute 3 numeric
7  @attribute 4 numeric
8  @attribute 5 numeric
9  @attribute 6 numeric
10 @attribute 7 numeric
11 @attribute 8 numeric
12 @attribute 9 numeric
13 @attribute class {E,Sc}
14
15 @data
16 0,1,0,1,0,0,0,0,2,3,E
17 0,9,2,0,6,2,6,9,0,3,Sc
18 0,1,0,2,1,0,0,1,0,1,E
19 0,7,0,0,6,0,0,3,0,4,E
20 4,13,2,1,6,5,0,6,1,3,Sc
21 0,6,0,0,2,0,0,3,0,2,E
22 0,7,0,1,3,0,0,9,0,4,E
23 3,2,3,2,3,3,3,1,0,1,Sc
```

Figura 32 - Exemplo de conjunto de dados final no formato ARFF
Fonte: Autoria Própria.

[14] Mais informações em http://www.cs.waikato.ac.nz/ml/weka/arff.html

O formato do arquivo: A anotação `@relation` detona o nome (arbitrário) do conjunto de dados e o início das especificações dos atributos que estruturam o mesmo. Cada atributo é anotado com `@attribute` seguido de seu nome (arbitrário) e do seu tipo definido por um dos padrões do formato (`numeric`, `string` etc). Por convenção o atributo que representa a classe normalmente é especificado por último, isto porque, a API determina o último atributo como sendo a classe da instancia, quando a mesma não é especificada. A anotação `@data` indica o início dos dados, cada linha representa uma instancia composta pelos valores dos atributos especificados.

Os descritores obtidos por meio das ferramentas da Sessão 4.3 são padronizados num arquivo ARFF, este então, é fornecido para o algoritmo de agrupamento para dar sequência à transformação das imagens, utilizando o modelo *Bag-Of-Keypoints.* Foi utilizado a implementação `SimpleKMeans`[15] para realização da tarefa de agrupamento no processo de construção do modelo *Bag-of-Keypoints*, o mesmo se baseia no algoritmo *K-Means* descrito brevemente na Sessão 3.2.3. O algoritmo foi utilizado com seus valores padrões, e sua medida de distância foi mantida como sendo a distância euclidiana. O único parâmetro que foi alterado foi o número de *clusters*, que funciona como um ponto de variação do tamanho do 'vocabulário' do modelo *Bag-of-Keypoints*.

O tamanho do vocabulário usado para criação da nova representação das imagens para os experimentos foi de 50.

O conjunto de dados transformados é então fornecido a cada um dos cinco indutores escolhidos para avaliação neste trabalho.

O algoritmo *MultiLayer Perceptron*[16] (MLP) foi escolhido como implementação da abordagem por redes neurais. Os parâmetros do algoritmo não foram alterados, ou seja, a taxa de aprendizagem foi mantida a mesma (0.3) assim como o número de épocas de treinamento (500).

A implementação da abordagem por *Lógica Fuzzy* escolhida foi o FURIA, que ainda não está presente na versão utilizada mas pode ser adquirida pelo gerenciador de pacotes da interface da WEKA. O algoritmo também foi utilizado com seus valores padrões.

---

[15] Disponível no pacote weka.clusterers.SimpleKMeans
[16] Disponível no pacote weka.classifiers.functions.MultilayerPerceptron

Para a abordagem dos *k-vizinhos* mais próximos foi utilizado a implementação *IBk*[17] (KNN) cujos parâmetros também foram mantidos padrões, sendo o número de vizinhos (*k*) igual a 1, sendo que o algoritmo para busca do vizinho mais próximo mante-se o *LinearSearch*, ou seja, busca por força bruta.

Na abordagem por árvores foi escolhida a implementação Random Forest[18] (RF), utilizada com seus valores padrões para construção dos modelos, mantendo-se o número de árvores igual a 100.

O algoritmo SMO[19] foi escolhido como implementação das máquinas de vetores de suporte, e foi utilizado com seus valores padrões.

4.5 AVALIAÇÃO

Para realizar a avaliação dos modelos foi utilizado o módulo *Experimenter* da ferramenta WEKA. Com ele é possível realizar experimentos com diversos conjuntos de dados e algoritmos. Deste modo todos os conjuntos de dados transformados, construídos a partir do modelo *Bag-of*-Keypoints, e de diferentes descritores, podem ser fornecidos a *n* algoritmos indutores para construção de modelos de detecção.

Cada indutor constrói dois modelos de detecção de um objeto *x*, a partir de dois conjuntos de dados que foram construídos, um baseado em descritores SIFT e outro baseado em descritores SURF. Cada modelo, construído a partir de determinado indutor é avaliado por meio do método de validação cruzada com 10 *folds*. Ao final são geradas todas as medidas referentes a qualidade das classificações realizadas por cada modelo criado juntamente com a medida do desvio padrão para verificar variabilidade dos resultados que ocorre em cada *fold* em relação ao média retornada pela validação cruzada.O desvio padrão é dado pela equação (13):

$$std = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1}} \qquad \textbf{(13)}$$

Onde $\overline{x}$ representa a média da amostra, ou seja, o valor retornado pela validação cruzada.

[17] Disponível no pacote weka.classifiers.lazy.IBk
[18] Disponível no pacote weka.classifiers.trees.RandomForest
[19] Disponível no pacote weka.classifiers.functions.SMO

Para geração das curvas ROC, foi utilizado o módulo *Knowledge Flow*. O WEKA utiliza uma classe chamada `ThresholdCurve` de um método da classe `Evaluation` para gerar os pontos a serem plotados no gráfico, que foram desenhados a partir de um pacote do `JFreeChart`, que pode ser baixado pela própria WEKA.

4.6 ANÁLISE E PROJETO

Para facilitar e organizar a construção do componente de classificação de imagens, foi utilizada a linguagem de modelagem UML (*Unified Modelling Language*) por meio do *software Astah* versão *Community*[20]. O mesmo possui diversas ferramentas para modelagem gráfica, os quais são utilizados na área de Engenharia de *Software* para documentação.

Nas próximas sessões são especificados alguns dos artefatos que foram utilizados para documentação e construção do componente.

4.6.1 Requisitos do Sistema

Os requisitos funcionais descritos a seguir se referem as funcionalidades que se espera que o componente desempenhe, os mesmos foram especificados para documentação, testes e validação:

- F1: O componente deve ser capaz identificar características que sejam invariantes a escala, rotação, iluminação e perspectiva em imagens.
- F2: O componente deve ser capaz de dar um ponto de extensão para aceitar novos algoritmos de identificação de características.
- F3: O componente deve ser capaz de transformar as características detectadas em formato descritor.

[20] Disponível em http://astah.net/editions/community

- F4: O componente deve ser capaz de dar um ponto de extensão para novos algoritmos de descrição de características.
- F5: O componente deve ser capaz de transformar as imagens numa representação compatível com a tabela atributo-valor.
- F6: O componente deve ser capaz de dar um ponto de extensão para a implementação de algoritmos de aprendizagem de máquina.
- F7: O componente deve conseguir identificar objetos em imagens com base nos modelos construídos com uma taxa de acurácia significativa.

4.6.2 Diagrama de Componentes

Na Figura 33 é ilustrado o diagrama de componentes que apresenta uma visão da estrutura física e específica os componentes das quais o módulo de classificação de imagem é dependente.

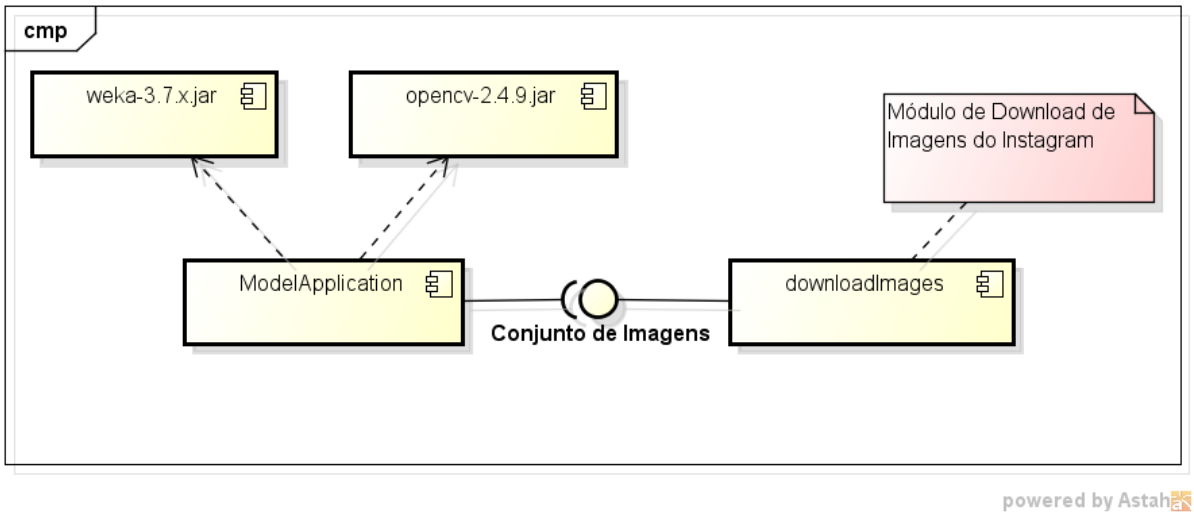


Figura 33 - Diagrama de componentes.
Fonte: Autoria Própria.

4.6.3 Diagrama de Classes

A modelagem do diagrama de classes propicia uma visão geral da estrutura e relação entre as classes que foram implementadas, como pode ser visto na Figura 34. O intuito é fornecer uma base para o entendimento de todo o projeto e de como ele foi construído.

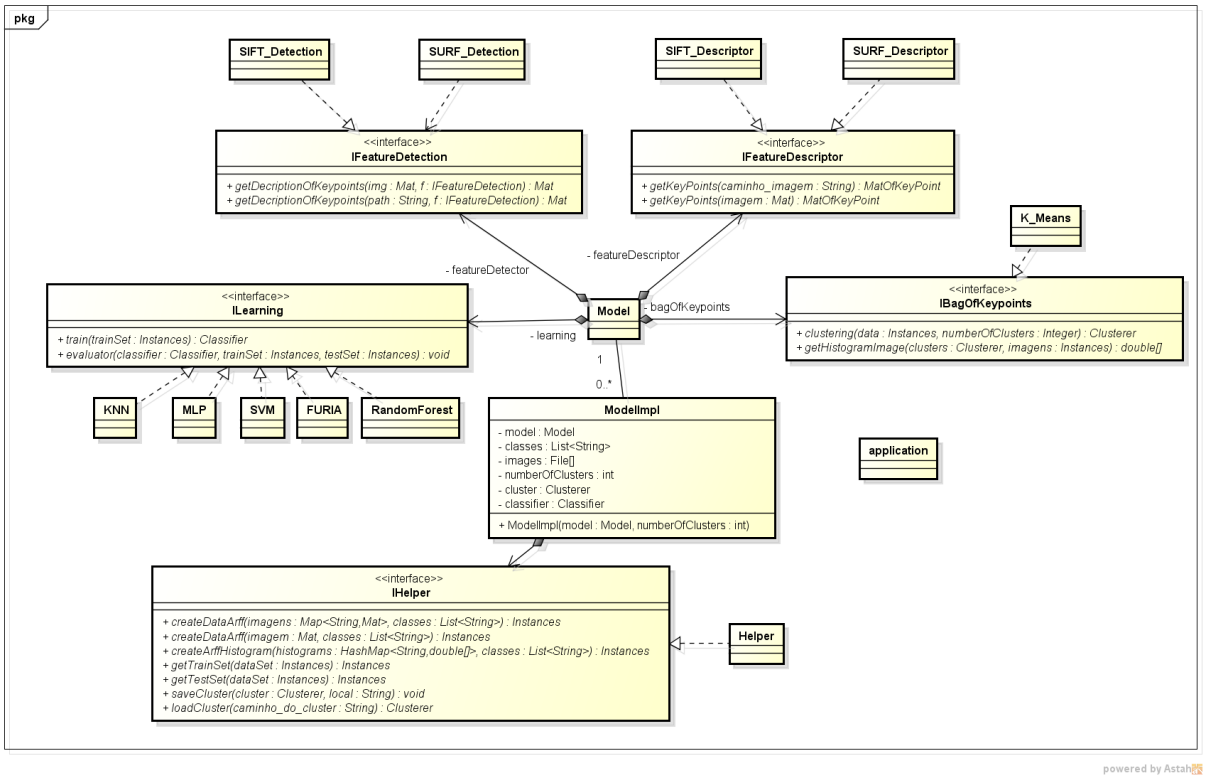


Figura 34 - Diagrama de classes.
Fonte: Autoria Própria.

5 RESULTADOS DOS CENÁRIOS EXPERIMENTAIS

A seguir serão descritos cenários nos quais os modelos descritos foram aplicados.

5.1 DETECÇÃO DE SAXOFONE

O primeiro cenário foi construído considerando um conjunto de dados pequeno para realizar a detecção de saxofones em imagens. O conjunto de treinamento fornecido para os indutores é composto por imagens da categoria *saxophone* presente no conjunto de dados *Caltech 101*. A distribuição de exemplos positivos e negativos pode ser vista na Tabela 4 os quais totalizam 86 exemplos.

Tabela 4 - Imagens de treinamento para detecção de saxofone.

Classe	Número de exemplos
comSax	40
semSax	46
Total	86

Fonte: Autoria Própria.

Na Tabela 5 é apresentada informações a respeito do custo computacional em termos de características extraídas e tempo computacional que foram gastos para transformar o conjunto de imagens. É possível notar que a implementação utilizada do algoritmo SURF detecta um número muito maior de características do que o algoritmo SIFT, porém tal diferença, não foi suficiente para gerar também uma diferença no tempo de transformação como um todo, especificamente na etapa de construção do *cluster*.

Tabela 5 - Custo e tempo computacional para transformação das imagens

Detector	Num. Carac. Descritas	Tempo aprox. *BoK*
SIFT	136241	1 hora
SURF	226858	1 hora

Fonte: Autoria Própria.

Os resultados das *performance* e qualidade obtidos no cenário experimental para detecção de saxofones pode ser observado na Tabela 6, nela estão dispostos os

valores obtidos por meio da 10-*fold* validação cruzada, das medidas de avaliação, juntamente com o desvio padrão (*std*) em relação aos *folds,* para cada Indutor treinado com base em um descritor (SIFT, SURF). Os melhores resultados estão destacados.

Tabela 6 - Desempenho e qualidade dos indutores para detecção de saxofone.

Indutor	Descritor	Acurácia (*std*)	Precision (*std*)	Recall (*std*)	MCC (*std*)
RF	SIFT	90.69 (*8.91*)	0.86 (*0.13*)	0.98 (*0.08*)	0.83 (*0.16*)
	SURF	96.53 (*5.60*)	0.96 (*0.08*)	0.98 (*0.08*)	0.94 (*0.10*)
FURIA	SIFT	83.75 (*14.60*)	0.81 (*0.17*)	0.90 (*0.13*)	0.69 (*0.28*)
	SURF	93.19 (*10.81*)	0.96 (*0.08*)	0.90 (*0.24*)	0.88 (*0.20*)
KNN	SIFT	90.83 (*11.86*)	0.86 (*0.16*)	1.00 (*0.00*)	0.85 (*0.19*)
	SURF	**98.89 (*3.51*)**	**0.98 (*0.06*)**	**1.00 (*0.00*)**	**0.98 (*0.06*)**
MLP	SIFT	88.75 (*11.72*)	0.86 (*0.16*)	0.95 (*0.11*)	0.80 (*0.20*)
	SURF	95.42 (*5.93*)	0.94 (*0.10*)	0.98 (*0.08*)	0.92 (*0.11*)
SMO	SIFT	87.08 (*13.05*)	0.81 (*0.16*)	1.00 (*0.00*)	0.78 (*0.21*)
	SURF	96.39 (*8.29*)	0.95 (*0.12*)	1.00 (*0.00*)	0.94 (*0.14*)

Fonte: Autoria Própria.

É possível observar pelos resultados apresentados, que os modelos criados com base em SURF geraram de modo significativo, melhores resultados de acurácia do que aqueles que foram gerados a partir da descrição SIFT. Dentre os indutores, o classificador KNN obteve os melhores resultados independentemente do descritor, e sua combinação com o descritor SURF obteve uma taxa de acurácia superior a 98%.

Sendo que o *recall* do modelo construído do indutor KNN com base em SURF, foi de 1.0, esta medida avalia a precisão do classificador de acertar as classes positivas, ou seja, o classificador acertou todos os exemplos que possuíam o objeto saxofone na imagem, os erros cometidos pelo classificador como pode ser analisada pela medida *Precision* indica que o classificador acabou classificando instâncias negativas como sendo positivas.

O valor MCC obtido pelo classificador KNN (0.98) indica que a qualidade de sua classificação foi quase perfeita. É possível perceber que o classificador também obteve os menores valores de desvio padrão, ou seja, os resultados diferem pouco entre as iterações da validação cruzada.

Na Figura 35 são apresentadas as curvas ROC para cada indutor construindo com base em descritor SIFT (Figura 35a) e descritor SURF (Figura 35b). Os gráficos apresentam uma visualização geral do desempenho de cada indutor. É possível observar que todos os classificadores se saíram melhor que a aleatoriedade, tendo suas

curvas acima da diagonal representada pela reta que parte de (0,0) até (1,1). Também observa-se que as curvas dos indutores construídos com base em SURF se aproximam mais do canto superior esquerdo, onde se encontra o ponto idealizado para o desempenho do classificador, ou seja, alta taxa de True Positive (verdadeiros positivos) e baixa taxa de falsos alarmes (False Positive). No caso das curvas observadas no gráfico construído com base em SIFT, observa-se que em alguns pontos alguns indutores se saem melhores do que outros, por exemplo, o indutor tende a ter desempenho melhor que o indutor KNN (IBk), mas em determinado ponto, o último acaba sobressaindo.

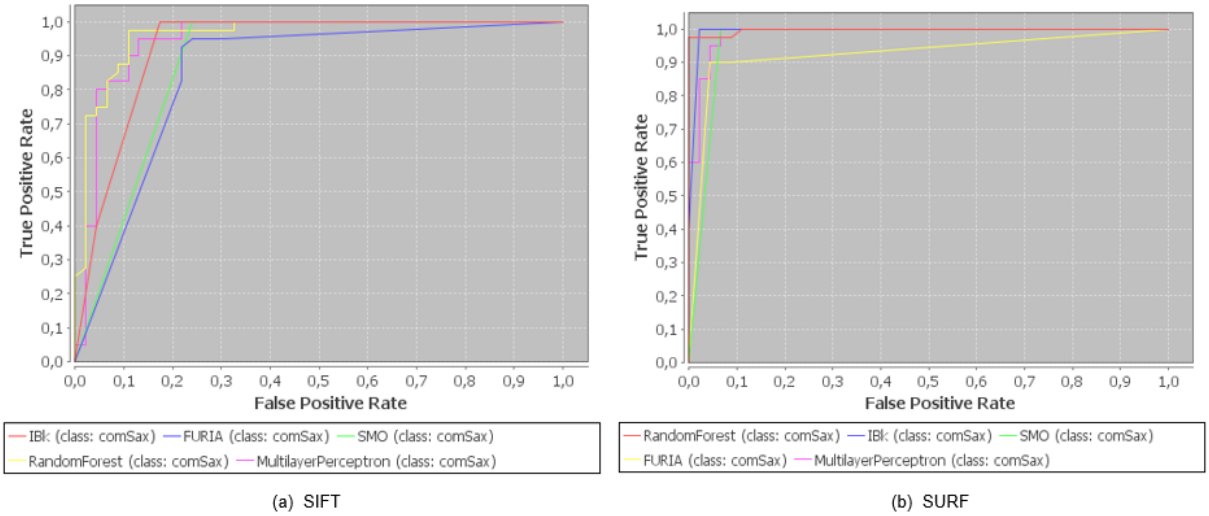


Figura 35 - Curvas ROC para detecção de saxofones.
Fonte: Autoria Própria.

Os valores da Área sobre a Curva ROC (AUC) são apresentadas na Tabela 7. Os valores demonstram novamente que todos os indutores conseguiram desempenhos melhores que o aleatório (AUC>0.5).

Tabela 7 - Áreas sobre as curvas ROC para detecção de saxofone.

Indutor	RF		FURIA		KNN		MLP		SMO	
Descritor	SIFT	SURF	SIFT	SURF	SIFT	SURF	SIFT	SURF	SIFT	SURF
AUC	0.96	0.99	0.85	0.93	0.92	0.99	0.95	0.98	0.88	0.97

Fonte: Autoria Própria.

Os indutores KNN e RF obtiveram um desempenho igual, segundo os valores obtidos, o que corrobora os valores obtidos com outras medidas, onde os dois indutores também obtiveram valores similares.

De modo geral, todos os modelos construídos obtiveram bons resultados.

5.2 DETECÇÃO DE CARRO

O segundo cenário experimental foi construído para detecção de carro considerando uma base de dados bem maior para o treinamento. As imagens contendo carros e que serviram como referência positiva aos indutores foram retiradas de um subconjunto do *GRAZ-02 database*, cerca de 420 exemplos da categoria carros, que possui imagens contendo carros em várias perspectivas e com oclusões. A distribuições de exemplos positivos e negativos pode ser conferida na Tabela 8.

Tabela 8 - Imagens de treinamento para detecção de carro.

Classe	Número de exemplos
comCarro	420
semCarro	377
Total	797

Fonte: Autoria Própria.

Na Tabela 9 contém as informações a respeito do número de características extraídas por cada detector e o tempo computacional que foram gastos para transformar o conjunto de imagens. Neste cenário também é possível notar que a implementação utilizada do algoritmo SURF detecta um número muito maior de características do que o algoritmo SIFT, contudo, neste cenário, o número influenciou de maneira muito significativa no tempo gasto para que o conjunto de dados fosse transformado.

Tabela 9 - Custo computacional e de tempo para detecção de carro.

Detector	Num. Carac. Descritas	Tempo aprox. *BoK*
SIFT	1776561	2 dias
SURF	2840960	3 dias

Fonte: Autoria Própria.

Devido ao grande número de características detectadas pelo algoritmo SURF, são gerados muito mais elementos para serem agrupados, aumentando assim o tempo para criação do *cluster*.

Os resultados do desempenho e qualidade dos modelos neste cenário são apresentados na Tabela 10, onde estão dispostos os valores obtidos por meio da 10-*fold* validação cruzada, das medidas de avaliação, juntamente com o desvio padrão (*std*) em relação aos *folds,* para cada Indutor treinado com base em um descritor (SIFT, SURF). O melhor modelo está destacado.

Tabela 10 - Desempenho e qualidade dos indutores para detecção de carro.

Indutor	Descritor	Acurácia (*std*)	Precision (*std*)	Recall (*std*)	MCC (*std*)
RF	SIFT	88.82 (*2.51*)	0.86 (*0.03*)	0.94 (*0.03*)	0.78 (*0.05*)
	SURF	**94.09 (*2.24*)**	**0.92 (*0.02*)**	**0.98 (*0.02*)**	**0.88 (*0.05*)**
FURIA	SIFT	84.80 (*4.45*)	0.84 (*0.05*)	0.88 (*0.06*)	0.70 (*0.09*)
	SURF	90.58 (*2.32*)	0.89 (*0.02*)	0.94 (*0.04*)	0.81 (*0.05*)
KNN	SIFT	85.17 (*2.96*)	0.81 (*0.04*)	0.93 (*0.01*)	0.71 (*0.05*)
	SURF	91.21 (*2.43*)	0.88 (*0.03*)	0.96 (*0.03*)	0.83 (*0.05*)
MLP	SIFT	88.45 (*3.71*)	0.84 (*0.04*)	0.96 (*0.03*)	0.78 (*0.07*)
	SURF	93.46 (*3.14*)	0.90 (*0.05*)	0.98 (*0.03*)	0.87 (*0.06*)
SMO	SIFT	86.93 (*3.21*)	0.81 (*0.03*)	0.98 (*0.03*)	0.75 (*0.06*)
	SURF	92.34 (*2.65*)	0.88 (*0.04*)	0.99 (*0.01*)	0.85 (*0.05*)

Fonte: Autoria Própria.

Da mesma forma que no cenário de detecção de saxofones, os modelos que foram construídos com base em SURF obtiveram resultados melhores. Dentre os modelos, o que apresentou melhores resultados de acurácia foi aquele construído com o indutor Random Forest (RF), sendo que sua combinação com o descritor SURF obteve uma taxa superior a 94%.

A medida *Recall* do indutor RF com base em SURF (0.98) indica que há grande eficiência em acertar as classes positivas, contudo o indutor que obteve melhor resultado nesta medida foi a implementação das máquinas vetores de suporte (SMO) com base em SURF, com um valor de *recall* quase perfeito (0.99), contudo o valor de *Precision* para o mesmo modelo indica uma pequena tendência do mesmo classificar classes como sendo positivas, e por outro lado, o indutor RF+SURF, obteve um valor de *Precision* superior, também indicando eficiência significativa do modelo em reconhecer a classe de interesse.

Todos os modelos obtiveram valores altos da medida MCC, o que indica que os mesmos possuem uma boa qualidade de classificação, pois se aproximam do valor positivo +1. Devido os valores das outras medidas influenciarem no cálculo do MCC, tem-se portanto que o indutor RF+SURF também obteve o melhor resultado (0.88). É possível perceber que o modelo obteve também os menores valores de desvio padrão, ou seja, os resultados diferente pouco entre as interações da validação cruzada.

As curvas ROC para cada modelo são apresentadas na Figura 36, onde as curvas dos modelos construídos com base em SIFT são apresentadas na Figura 36a, e os construídos com base em SURF são apresentadas na Figura 36b. É possível observar que todos os modelos se mostraram melhores que a aleatoriedade, tendo suas curvas desenhadas acima da diagonal. Visualmente as curvas que se destacam e se aproximam mais do ponto ideal, são as dos modelos construídos com os indutores *Random Forest* (RF) e *MultiLayer Perceptron* (MLP).

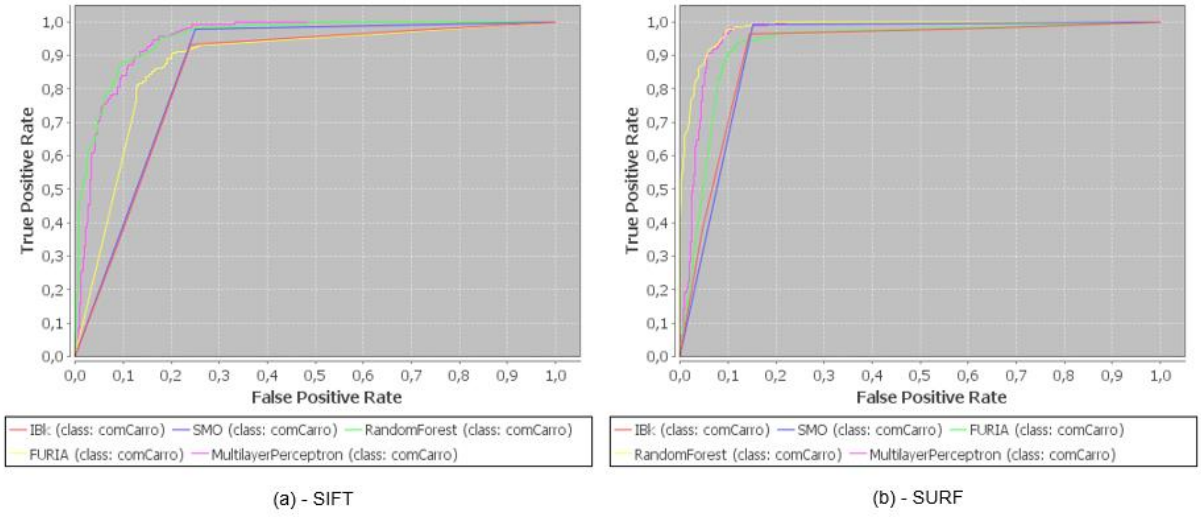


Figura 36 - Curvas ROC para detecção de carro.
Fonte: Autoria Própria.

Os valores da Área sobre a Curva ROC (AUC) são apresentadas na Tabela 11. Os valores demonstram novamente que todos os indutores conseguiram desempenhos melhores que o aleatório (AUC>0.5).

O modelo construído com o indutor RF obtive o valor de 0.98 que corrobora com os valores obtidos com outras medidas e com a visualização a partir de sua curva ROC. De modo geral, todos os modelos construídos obtiveram bons resultados.

Tabela 11 - Áreas sobre as curvas ROC para detecção de carro.

Indutor	RF		FURIA		KNN		MLP		SMO	
Descritor	SIFT	**SURF**	SIFT	SURF	SIFT	SURF	SIFT	SURF	SIFT	SURF
AUC	0.95	**0.98**	0.88	0.93	0.85	0.91	0.95	0.97	0.86	0.92

Fonte: Autoria Própria.

5.3 CONSIDERAÇÕES SOBRE O CENÁRIO EXPERIMENTAL

Os resultados obtidos com os modelos construídos para o cenário experimental de detecção de saxofones se demonstraram promissores, com taxas significativas de acertos em todas as combinações de Descritores e Indutores, mesmo com uma base de treinamento relativamente pequena.

Da mesma forma os modelos construídos para detecção de carros obtiveram resultados significativos, porém com um desempenho menor do que o primeiro cenário, mesmo tendo eles sido construídos com uma base de treinamento significativamente maior.

Algumas hipóteses são criadas a partir da análise dos resultados, os quais não foram testadas nesse trabalho pois estenderiam o escopo do mesmo. É possível observar um melhor desempenho dos algoritmos indutores quando treinados com base em descritor SURF. Esta alta taxa pode estar associada ao número de características detectadas por cada algoritmo com base em seu Detector, o que gera a descrição de mais padrões que são fornecidos posteriormente aos indutores, que acarreta na construção de um modelo mais robusto. Não foi realizada a combinação de detector/descritor, portanto, tem-se a possibilidade de verificar a hipótese em trabalhos futuros.

6 APLICAÇÃO DOS MODELOS EM IMAGENS DO INSTAGRAM

Tendo os modelos sido construídos e avaliados considerou-se utilizar os dois modelos que obtiveram os melhores desempenhos em cada cenário para realizar teste em um cenário real, para verificar eficiência. Utilizando o módulo para download de imagens do *Instagram*, foram adquiridas imagens por meio de *tags* especificadas pelo nome do objeto para haver maior probabilidade de que as imagens retornadas contivessem algum dos mesmos. Os resultados desse cenário se encontram a seguir.

6.1 DETECÇÃO DE SAXOFONE

Para o teste de detecção de saxofone foram adquiridas 20 imagens, sendo elas visualmente categorizadas com contendo ou não o objeto. Deste total foram separados 7 exemplos contendo saxofone e 7 não contendo saxofone, totalizando 14 exemplos para o teste. Os dois modelos que obtiveram os melhores resultados no cenário experimental foram os modelos construídos pelos indutores KNN e RF, ambos com base em SURF.

Na Tabela 12 se encontra o resultado da taxa de acertos do teste no cenário real. É possível notar que acurácia em relação ao cenário experimental caiu drasticamente, o algoritmo KNN manteve-se com o melhor desempenho nesse cenário.

Tabela 12 - Taxa de acertos no cenário real de detecção de saxofone

Modelo	Acurácia no Teste
SURF-KNN	78.57%
SURF-RF	64.28%

Fonte: Autoria Própria.

Na Figura 37 são ilustrados os exemplos de acerto e erro do modelo que se saiu melhor no cenário de teste, no caso o modelo construído com o indutor KNN. As imagens ilustradas são os exemplos positivos do conjunto de teste, ou seja, imagens que visualmente possuíam saxofone. As imagens que o indutor acertou estão destacadas com um retângulo verde, e as que possuem um retângulo vermelho ao redor,

são imagens que o indutor classificou como não contendo o objeto. Os exemplos negativos não foram ilustrados pois ambos os modelos acertaram todos os exemplos negativos, o que nota uma tendência dos mesmos em classificar as instâncias como sendo negativas.

Figura 37 - Aplicação do modelo em cenário real de detecção de saxofone.
Fonte: Autoria Própria.

6.2 DETECÇÃO DE CARRO

Da mesma forma que o teste para detecção de saxofone, para o teste de detecção de carro foram adquiridas 20 imagens, que foram visualmente categorizadas com contendo ou não, carro. Deste total foram separados 8 exemplos contendo carro e 7 não contendo carro, totalizando 15 exemplos para o teste. Os dois modelos que obtiveram os melhores resultados no cenário experimental foram os modelos construídos pelos indutores RF e MLP ambos com base em SURF.

Os resultados obtidos no teste do cenário real podem ser observados na Tabela 13. O modelo que obteve os melhores resultados no cenário experimental (SURF-RF) obteve uma taxa de acertos muito inferior ao que era esperado, cerca de 33%, sendo

que das 8 instâncias contendo carro, 5 ele classificou como não contendo e das 7 instâncias que não continham carro, ele acabou classificando como contendo.

Tabela 13 - Taxa de acertos no cenário real de detecção de carro

Modelo	Acurácia no Teste
SURF-RF	33%
SURF-MLP	66%

Fonte: Autoria Própria.

O modelo SURF-MLP, apesar de apresentar resultados abaixo do que se esperava quando comparado aos resultados no cenário experimental, obteve uma taxa de acerto um pouco melhor, cerca de 66%. As imagens positivas utilizadas no teste podem ser observadas na Figura 38, nela estão destacados os acertos e erros do modelo MLP (que obteve o melhor resultado neste cenário). As imagens que estão destacadas com um retângulo verde são as instâncias que ele corretamente classificou, e as que possuem um retângulo vermelho a volta são as que ele incorretamente classificou, ou seja, das 8 contendo carro, ele classificou 3 como não contendo.

Figura 38 - Aplicação no cenário real de detecção de carro
Fonte: Autoria Própria.

Os erros dos exemplos negativos são ilustrados na Figura 39, que são imagens que não continham carro e acabaram sendo classificadas como contendo o objeto.

Figura 39 - Imagens que não contem carro que foram classificadas erradas
Fonte: Autoria Própria.

6.3 CONSIDERAÇÕES SOBRE O CENÁRIO REAL

As imagens adquiridas de redes sociais podem vir em diversas resoluções e sobre as mais diferentes formas, por exemplo, a iluminação do cenário pode ser baixa ou a imagem pode ter sofrido rotação e mudança na escala. Somando esses detalhes ao fato de quando é analisado especificamente os objetos que podem estar contidos nelas, os mesmos, podem estar oclusos, ou com muita informação a volta, o que aumenta complexidade para um algoritmo identificar os padrões.

Os resultados obtidos no cenário real revelam pouca eficiência dos modelos construídos pelo módulo de classificação para detectar os objetos. Algumas considerações hipotéticas para essa pouca eficiência dos modelos foram levantadas.

Com relação ao modelo de detecção de saxofones, sua pouca eficiência pode advir do fato de poucos exemplos terem sido fornecidos para treinamento, sendo assim, o indutor foi incapaz de construir um modelo robusto para detectar os objetos nas mais variadas situações.

Contudo, fornecer uma grande quantidade de exemplos de treinamento para um indutor, não significa que ele será capaz de construir um modelo significativamente confiável. Como pode ser visto no caso do segundo cenário, onde o conjunto de treinamento fornecido para os indutores foi consideravelmente maior do que no primeiro cenário. Quando analisado o conjunto de dados que foi utilizado para detecção de carro, nota-se pouca diversidade de ‘situações’ onde um carro está contido. Portanto deve haver além de quantidade, uma boa variedade de imagens.

Como exemplo, considerando o conjunto de dados de treinamento para detecção de carro, nota-se que nas imagens não há a presença de pessoas, o que pode levar os indutores a identificar o padrão tendencioso de que pessoas na imagem significam a ausência de carro.

Outra consideração importante diz respeito a resolução das imagens. Os exemplos negativos que foram fornecidos aos indutores para a criação dos modelos são provenientes de uma rede social (Flickr), e as mesmas possuem altas resoluções. Em contrapartida as imagens dos conjuntos de dados utilizados como exemplos positivos para treinamento, têm pouca variação na resolução, das quais são retiradas um número baixo de características quando comparadas aos exemplos negativos. Deste modo, uma imagem retirada de uma mídia social que tenha alta resolução, acarretará na extração de um número muito maior de características, que por sua vez, levará a construção de uma nova representação por meio do modelo *Bag-of-Keypoints* com valores dos atributos muito altos e discrepantes dos padrões que foram fornecidos aos indutores para construção dos modelos, ocasionando o erro, que se evidencia sob a forma tendenciosa dos indutores classificarem imagens com altas resoluções como não contendo o objeto de interesse.

7 CONSIDERAÇÕES FINAIS

O componente implementando atendeu as especificações definidas pelos requisitos funcionais, fornecendo pontos de extensão para a implementação de novos algoritmos de detecção de características e descrição, assim como algoritmos de aprendizagem de máquina. O componente também realiza a transformação das imagens, mapeando os dados extraídos das mesmas para o formato atributo-valor utilizando o modelo *Bag-of-Keypoints* que é reconhecido pelo meio cientifico e bastante aplicado na área de visão computacional.

Os resultados nos cenários experimentais indicaram uma eficiência significativa dos modelos construídos com base no algoritmo de detecção e descrição SURF, tendo taxas de acerto de mais de 98% no cenário de detecção de saxofone, e mais de 94% no cenário de detecção de carros. Contudo, no cenário de aplicação dos modelos construídos em imagens provenientes de redes sociais, verificou-se quedas significativas de desempenho.

Deste modo, o requisito referente a significativa taxa de acurácia esperada dos modelos construídos ficou inconclusivo pela falta de mais experimentos para verificar as hipóteses levantadas acerca dos resultados obtidos. Contudo o componente funciona da maneira esperada, realizando as tarefas que vão das transformações das imagens em dados até a construção de modelos para classificação.

Apesar de não terem taxas de acerto significativamente altos no cenário real, os modelos possuem desempenhos consideráveis que podem ser melhorados mediante mais estudo e testes dos parâmetros envolvidos na construções dos mesmos, por meio de novos experimentos, assim como o refinamento das bases de dados de treinamento que foram fornecidas aos indutores, e que tem grande impacto em sua qualidade final.

7.1 TRABALHOS FUTUROS

Para trabalhos futuros pretende-se verificar hipóteses por meio de experimentos que não foram realizados neste trabalho:

- Construir modelos para a detecção de outros objetos, visando por exemplo, detectar armas.
- Realizar experimentos com a combinação de detector de características e descritor.
- Avaliar outros algoritmos de detecção e descrição tais como: FAST, ORB (*oriented BRIEF*) e FREAK (*Fast Retina Keypoint*)
- Construir bases de dados mais completas, diversificadas e com resoluções padronizadas para serem fornecidas como conjunto de treinamento/conhecimento.
- Otimizar os algoritmos de aprendizagem de máquina buscando melhores desempenhos.
- Utilizar técnicas de *boosting* buscando melhorar o desempenho dos indutores.
- Realizar a construção de uma interface para utilização do componente.

REFERÊNCIAS

ACHARYA, T.; RAY, A. K. **Image processing:** principles and applications. [S.l.]: John Wiley & Sons, 2005.

ALAHI, ; ORTIZ, ; VANDERGHEY, P. Freak: Fast retina keypoint. **Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on**, p. 510--517, 2012.

ALMEIDA, J.; TORRES, R. D. S.; GOLDENSTEIN, S. SIFT applied to CBIR. **Revista de Sistemas de Informacao da FSMA n**, v. 4, p. 41-48, 2009.

ALPAYDIN, E. **Introduction to Machine Learning**. 2. ed. London, England: The MIT Press, 2010.

ALVES, G. T. M.; GATTASS, M.; CARVALHO, P. C. P. **Um Estudo das Técnicas de Obtenção de Forma a partir de Estéreo e Luz Estruturada para Engenharia**. Rio de Janeiro: Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, 2005. 88p p.

BALLARD, D. H.; BROWN, C. M. **Computer Vision**. Englewood Cliffs ,: Prentice Halll, 1982.

BARANAUSKAS, J. A. **Aprendizado de Máquina, Conceitos e Definições: Notas de Aula**. Departamento de Física e Matemática – FFCLRP-USP. [S.l.], p. 20. 2007.

BARANAUSKAS, J. A.; MONARD, M. C. **Reviewing some machine learning concepts and methods**. ICMC - USP. São Carlos. 2000. (102).

BARBOSA,. **Efficient Database Image Search**. Faculdade de Engenharia da Universidade do Porto. Porto, p. 18. 2014.

BAY, Herbert; ESS , Andreas ; TUYTELAARS, Tinne ; VAN GOOL, Luc. Speeded-up robust features (SURF). **Computer vision and image understanding**, v. 110, n. 3, p. 346--359, 2008.

BELO, F. A. W. **Desenvolvimento de Algoritmos de Exploração e Mapeamento Visual**. Rio de Janeiro: Dissertação de Mestrado - Departamento de Engenharia Elétrica, 2006. 276 p.

BEZDEK, James C.; KELLER, James; KRISNAPURAM, Raghu; PAL, Nikhil R.. **Fuzzy Models and Algorithms for Pattern Recognition and Image Processing**. [S.l.]: Springer, v. 4, 2005.

BOSCH, A.; MUÑOZ, X.; MARTÍ, R. Which is the best way to organize/classify images by content? **Image and vision computing**, v. 25, n. 6, p. 778-791, 2007.

BOURIDANE, A. **Imaging for Forensics and Security:** From Theory to Practice. [S.l.]: Springer, v. 106, 2009.

BRADSKI, G.; KAEHLER, A. **Learning OpenCV:** Computer vision with the OpenCV library. [S.l.]: O'Reilly Media, Inc., 2008.

BRAMER, M. **Principles of Data Mining**. [S.l.]: Springer, v. 180, 2007.

BREIMAN, L. Random forests. **Machine learning**, v. 45, n. 1, p. 5-32, 2001.

BROWN, M.; LOWE, D. G. Automatic panoramic image stitching using invariant features. **International journal of computer vision**, 74, n. 1, 2007. 59-73.

BUENO, L. M. **Análise de descritores locais de imagens no contexto de detecção de semi-réplicas.** Campinas: Universidade Estadual de Campinas, 2011.

CARVALHO, I. A. D. **Classificação de imagens de pornografia e pornografia infantil utilizando recuperação de imagens baseada em conteúdo.** Brasília: Universidade de Brasília, 2012.

CIOS, Krzysztof J.; PEDRYCZ, Witold; SWINIARSKI, Roman W.; KURGAN, Lukasz A.. **Data Mining:** A Knowledge Discovery Approach. [S.l.]: Springer, 1998.

CIPOLLA, R.; BATTIATO, S.; FARINELLA, G. M. **Machine Learning for Computer Vision**. [S.l.]: Springer Publishing Company, Incorporated, 2012.

CLARKE, B.; FOKOUE, E.; ZHANG, H. H. **Principles and theory for data mining and machine learning**. [S.l.]: Springer Science & Business Media, 2009.

CONCI, A.; AZEVEDO, E.; LETA, F. R. **Computação Gráfica:** Teoria e Prática. Rio de Janeiro: Elsevier, v. 2, 2008. 407 p.

CSURKA, Gabriella; DANCE, Christopher; FAN, Lixin; WILLAMOWSKI, Jutta; BRAY, Cédric. Visual categorization with bags of keypoints. **Workshop on statistical learning in computer vision, ECCV**, p. 1-2, 2004.

DE ANDRADE, F. D. S. P. **Combinação de descritores locais e globais para recuperação de imagens e vídeos por conteúdo**. Biblioteca Digital da Unicamp. [S.l.]. 2012.

DE ARAÚJO, S. A. **Casamento de padrões em imagens digitais livre de segmentação e invariante sob transformaçõess de similaridade**. Universidade de São Paulo. [S.l.]. 2009.

DE LA CALLEJA, J.; FUENTES, O. Automated classification of galaxy images. **Knowledge-Based Intelligent Information and Engineering Systems**, p. 411-418, 2004.

DESELAERS, T. Features for image retrieval. **Rheinisch-Westfalische Technische Hochschule, Technical Report, Aachen**, 2003.

DOUGHERTY, G. **Digital image processing for medical applications**. [S.l.]: Cambridge University Press, 2009.

DOUGHERTY, G. **Pattern Recognition and Classification:** An Introduction. Camarillo, CA, USA: Springer, 2013.

FAWCETT, T. ROC graphs: Notes and practical considerations for researchers. **Machine learning**, v. 31, p. 1-38, 2004.

FEI-FEI, L.; FERGUS, R.; PERONA, P. Learning generative visual models. **CVPR 2004, Workshop on Generative-Model**, 2004.

FLITTON, ; BRECKON, T. P.; MEGHERBI, N. A 3D extension to cortex like mechanisms for 3D object class recognition. **Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on**, p. 3634--3641, 2012.

FLITTON, G. T.; BRECKON, T. P.; BOUALLAGU, N. M. Object Recognition using 3D SIFT in Complex CT Volumes. **BMVC**, p. 1-12, 2010.

GHELLERE, J.S.; HOFFMANN, A.B.G.; SANTANA, P.; METZ, J. Classificação de galáxias usando descritores locais e Bag-Of-Keypoints. In: **MEDIANEIRA IN TECHNOLOGY (MEDITEC), 6**., 2015, Medianeira. Anais... Medianeira: UTFPR, 2015.

GONZÁLES, G. L. G.; MEGGIOLARO, M. A. **Aplicação da Técnica SIFT para Determinação de Campos de Deformações de Materiais usando Visão Computacional**. PUC-Rio. [S.l.], p. 109p. 2010.

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**. 2. ed. [S.l.]: Prentice Hall, 2002.

GRAUMAN, K.; LEIBE, B. **Visual object recognition**. [S.l.]: Morgan & Claypool Publishers, 2011.

HALL, Mark; FRANK, Eibe; HOLMES, Geoffrey; PFAHRINGER, Bernhard; REUTEMANN, Peter; WITTEN, Ian H. The WEKA data mining software: an update. **ACM SIGKDD explorations newsletter**, v. 11, p. 10-18, 2009.

HÜHN, J.; HÜLLERMEIER, E. FURIA: An Algorithm For Unordered Fuzzy Rule Induction. **Data Mining Knowl. Discovery**, v. 19, p. 293-319, 2009.

JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. Data clustering: a review. **ACM computing surveys (CSUR)**, v. 31, p. 264-323, 1999.

KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. **appers in the international joint conference on Artificial intelligence (IJCAI)**, 1995. 1137-1145.

KONONENKO, I.; KUKAR, M. **Machine learning and data mining:** Introduction to Principles and Algorithms. [S.l.]: Horwood Publishing, 2007.

LAGAREIRO, A. C. Fotografias tipo “selfie”: Uma forma de expressão ou a manifestação de um comportamento preocupante? Disponivel em: <http://www.pucsp.br/nppi/coluna_eletronica/2014/selfie-janeiro-2014.pdf>. Acesso em: 17 set 2014.

LAVESSON, N. **Evaluation and analysis of supervised learning algorithms and classifiers**. Karlskrona, Sweden: Blekinge Institute of Technology, 2006.

LECUN, Y.; HUANG, F. J.; BOTTOU, L. Learning methods for generic object recognition with invariance to pose and lighting. **In: Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on.**, 2, 2014. II-97.

LEUTENEGGER, S. **Image Keypoint Detection, Description, and Matching**. AIRobots Summer School. [S.l.]. 2012.

LIBERMAN, F. **Classificação de imagens digitais por textura usando redes Neurais**. Rio Grande do Sul: Universidade Federal do Rio Grande do Sul. Instituto de Informática. Curso de Pós-Graduação em Ciência da Computação, 1997.

LISIN, Dimitri A; MATTAR, Marwan A; BLASCHKO, Matthew B; LEARNED-Miller, Erik G; BENFIELD, Mark C. Combining local and global image features for object class recognition. **Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on**, 2005. 47-47.

LORENA, A. C.; DE CARVALHO, A. C. **Introdução às Máquinas de Vetores Suporte**. Relatório Técnico do Instituto de Ciências Matemáticas e de Computação (USP/Sao Carlos). [S.l.]. 2003.

LOWE, D. G. Object recognition from local scale-invariant features. **Computer vision, 1999. The proceedings of the seventh IEEE international conference on**, v. 2, p. 1150--1157, 1999.

LOWE, D. G. Distinctive image features from scale-invariant keypoints. **International journal of computer vision**, v. 60, n. 2, p. 91--110, 2004.

METZ, J. **Interpretação de clusters gerados por algoritmos de clustering hierárquico**. Dissertação. Universidade de São Paulo. [S.l.]. 2006.

MITCHELL, T. M. **Machine Learning**. [S.l.]: WCB/McGral-Hill, 1997.

MONARD, M. C.; BARANAUSKAS, J. A. **Conceitos sobre Aprendizagem de Máquina:** Chapter 4. 1. ed. [S.l.]: [s.n.], v. 1 of Rezende (2003), 2003.

NILSSON, N. J. **Introduction to machine learning: An early draft of a proposed textbook**. Stanford University. Stanford , p. 201. 1996.

NIXON, M. S.; AGUADO, A. S. **Feature Extraction and Image Processing**. 1. ed. [S.l.]: Newnes, 2002.

OPELT, Andreas; PINZ, Axel; FUSSENEGGER, Michael; AUER, Peter. Generic object recognition with boosting. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, v. 28, p. 416-431, 2006.

PARK, U.; PANKANTI, ; JAIN, A. Fingerprint verification using SIFT features. **SPIE Defense and Security Symposium**, p. 69440K-69440K, 2008.

PARKER, C. An analysis of performance measures for binary classifiers. **Data Mining (ICDM), 2011 IEEE 11th International Conference on**, 2011. 517-526.

PAULA FILHO, P. L. **Processamento Digital de Imagens**. Notas de Aula. 2014.

PAULA FILHO, P. L. D. **Reconhecimento de espécies florestais através de imagens macroscópicas**. Tese (Doutorado) – Universidade Federal do Paraná. Curitiba. 2013.

PEDRINI, H.; SCHWARTZ, W. R. **Análise de Imagens Digitais:** Princípios, Algortimos e Aplicações. São Paulo: Thompson, 2008.

PENATTI, O. A. B. **Estudo comparativo de descritores para recuperação de imagens por conteudo na web**. Biblioteca Digital da Unicamp. [S.l.]. 2009.

POLAKOWSKI, William E; COURNOYER, Donald A; ROGERS, Steven K; DESIMIO, Martin P; RUCK, Dennis W; HOFFMEISTER, Jeffrey W; RAINES, Richard A. Computer-aided breast cancer detection and diagnosis of masses using difference of Gaussians and derivative-based feature saliency. **Medical Imaging, IEEE Transactions on**, v. 16, n. 6, p. 811-819, 1997.

PRATI, R.; BATISTA, G.; MONARD, M. Curvas ROC para avaliação de classificadores. **Revista IEEE América Latina**, v. 6, p. 215-222, 2008.

RAOUI, Y.; BOUYAKHF, E. H. A. D. M.; REGRAGUI, F. Global and local image descriptors for Content Based Image Retrieval and object recognition. **Applied Mathematical Sciences**, 5, 2011. 2109-2136.

ROMANO, R. A.; ARAGON, C. R.; DING, C. Supernova recognition using support vector machines. **Machine Learning and Applications, 2006. ICMLA'06. 5th International Conference on**, p. 77-82, 2006.

ROMBERG, Stefan; PUEYO, Lluis Garcia; LIENHART, Rainer; VAN ZWOL, Roelof. Scalable logo recognition in real-world images. **Proceedings of the 1st ACM International Conference on Multimedia Retrieval**, Trento, Italy, 2011. Disponivel em: <http://www.multimedia-computing.de/flickrlogos/>.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence:** A modern approach. 3ª. ed. [S.l.]: Prentice Hall Series in Artificial Intelligence, 1995.

SALTON, G.; MCGILL, M. J. **Introduction to Modern Information Retrieval**. New York, NY, USA: McGraw-Hill, Inc., 1986.

SCHMID, C.; ROGER, M.; BAUCKHAGE, C. Comparing and evaluating interest points. **Computer Vision, 1998. Sixth International Conference on**, p. 230-235, 1998.

SEBE, N; COHEN, IRA; GARG, ASHUTOSH; HUANG, THOMAS S. **Machine Learning in Computer Vision**. [S.l.]: Springer, 2005.

SIVIC, J.; ZISSERMAN, A. Video Google: A text retrieval approach to object matching in videos. **Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on**, p. 1470-1477, 2003.

STARCK, J.-L.; MURTAGH, F. **Astronomical Image and Data Analysis**. 2. ed. [S.l.]: Springer, 2006.

SZELISKI, R. **Computer Vision:** Algorithms and Applications. [S.l.]: Springer, 2010.

TAGLIAFERRI, R; LONGO, G; ANDREON, S; ZAGGIA, S; CAPUANO, N; GARGIULO, G. Astronomical object recognition by means of neural networks. **Neural Nets WIRN VIETRI-98**, p. 169-178, 1999.

THEODORIDIS, S.; KOUTROUMBAS, K. **Pattern Recognition**. 3. ed. San Diego: Elsevier, 2006. 837 p.

TIAN, D. P. A Review on Image Feature Extraction and Representation Techniques. **International Journal of Multimedia and Ubiquitous Engineering**, v. 8, n. 4, p. 385-396, 2013.

TUYTELAARS, T.; MIKOLAJCZYK, K. Local invariant feature detectors: a survey. **Foundations and Trends in Computer Graphics and Vision**, 3, 2008. 177-280.

WU, Xindong; KUMAR, Vipin; QUINLAN, J Ross; GHOSH, Joydeep; YANG, Qiang; MOTODA, Hiroshi; MCLACHLAN, Geoffrey J; NG, Angus; LIU, Bing; PHILIP, S Yu; outros. Top 10 algorithms in data mining. **Knowledge and Information Systems**, v. 14, p. 1-37, 2008.