UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE COMPUTAÇÃO
CURSO DE CIÊNCIA DA COMPUTAÇÃO

CARLOS CLAUDINEI SIMÕES

RECONHECIMENTO DE LOGOTIPOS USANDO DEEP LEARNING E RECUPERAÇÃO DE IMAGEM BASEADA EM CONTEÚDO

TRABALHO DE CONCLUSÃO DE CURSO

MEDIANEIRA

2018

CARLOS CLAUDINEI SIMÕES

RECONHECIMENTO DE LOGOTIPOS USANDO DEEP LEARNING E RECUPERAÇÃO DE IMAGEM BASEADA EM CONTEÚDO

Trabalho de Conclusão de Curso apresentado ao Departamento Acadêmico de Computação da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de "Bacharel em Computação".

Orientador: Prof. Dr. Pedro Luiz de Paula Filho

Co-orientador: Prof. Dr. Arnaldo Candido Junior

MEDIANEIRA

2018

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Diretoria de Graduação e Educação Profissional
Coordenação do Curso de Ciência da Computação

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

TERMO DE APROVAÇÃO

RECONHECIMENTO DE LOGOTIPOS USANDO DEEP LEARNING E RECUPERAÇÃO DE IMAGEM BASEADA EM CONTEÚDO

Por

CARLOS CLAUDINEI SIMÕES

Este Trabalho de Conclusão de Curso foi apresentado às 09:00h do dia 13 de junho de 2018 como requisito parcial para a obtenção do título de Bacharel no Curso de Ciência da Computação, da Universidade Tecnológica Federal do Paraná, Câmpus Medianeira. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

_____

Prof. Dr. Pedro Luiz de Paula Filho
UTFPR - Câmpus Medianeira

_____

Prof. Dr. Arnaldo Candido Junior
UTFPR - Câmpus Medianeira

_____

Prof. Dr. Paulo Lopes de Menezes
UTFPR - Câmpus Medianeira

_____

Prof. Msc. Jorge Aikes Junior
UTFPR - Câmpus Medianeira

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

RESUMO

SIMÕES, Carlos Claudinei. RECONHECIMENTO DE LOGOTIPOS USANDO DEEP LEARNING E RECUPERAÇÃO DE IMAGEM BASEADA EM CONTEÚDO. 57 f. Trabalho de Conclusão de Curso – Curso de Ciência da Computação, Universidade Tecnológica Federal do Paraná. Medianeira, 2018.

Com a expansão da quantidade de dados visuais gerados diariamente, surge a necessidade de sistemas capazes de realizar a classificação ou detecção dos mesmos de maneira automatizada. Neste trabalho é proposto um método alternativo para implementação de um sistema de Recuperação de Imagem Baseado em Conteúdo (CBIR), onde o processo de extração de características se da por meio de uma Rede Neural Convolucional (CNN) ao invés de se realizar a seleção e utilização de descritores específicos. O processo adotado se resume em realizar o treinamento de uma CNN, e posteriormente fazer algumas alterações na estrutura da mesma, removendo as camadas finais, assim sendo possível obter vetores com características referente às imagens, essas características obtidas são então utilizadas no CBIR. Foi possível alcançar excelentes resultados utilizando o método proposto, sendo que dentro da base de imagens para testes que contava com 20 imagens, foi obtida uma acurácia de 95% quando utilizado em uma tarefa de classificação.

Palavras-chave: reconhecimento de logotipos, extração de características, deep learning, cbir, cnn

ABSTRACT

SIMÕES, Carlos Claudinei. LOGO RECOGNITION WITH DEEP LEARNING AND CONTENT-BASED IMAGE RETRIEVAL. 57 f. Trabalho de Conclusão de Curso – Curso de Ciência da Computação, Universidade Tecnológica Federal do Paraná. Medianeira, 2018.

With the growth of the amount of visual data generated daily, there is a need for systems capable of classifying or detecting them in an automated way. This work proposes an alternative method for the implementation of a Content-Based Image Retrieval (CBIR) system, where the process of feature extraction is done through a Convolutional Neural Network (CNN) instead of picking specific descriptors. The process adopted consists of performing the training of a CNN, and then making some changes in the structure of the CNN, removing the final layers, this way it is possible to obtain feature vectors from the images, and then use these features in the CBIR system. It was possible to achieve excellent results using the proposed method, and within the base of images for tests that counted with 20 images, an accuracy of 95 % was obtained when used in a classification task.

Keywords: logo recognition, feature extraction, deep learning, cbir, cnn

AGRADECIMENTOS

Primeiramente e acima de tudo agradeço aos meus pais, Claudinei e Ivoneide, a todo o apoio, esforço e sacrifícios que fizeram ao longo desses anos para que eu pudesse estar aqui hoje finalizando este trabalho. Também aos meus irmãos Bruno e Thiago, que compartilharam de tudo isso.

Agradeço aos amigos, colegas e conhecidos, pelos bons momentos que proporcionaram ao longo dessa árdua jornada. Dentre todos os que conheci na Universidade, um salve para o Alex, Marcos, Gustavo, Gabriela, Eduardo e Marcelo. Um salve pro Felipe que além de colega de turma, já foi colega de estágio e agora de joguinho online. Um agradecimento para Julio, Matheus, Douglas, Lucas, Adriano, Aguinaldo, Gustavo e Crow, amigos que fiz jogando nos fins de semana e férias, e que alguns pude conhecer pessoalmente depois. E por fim um agradecimento grande igual eu, para a Valéria, que desde o inicio foi meu braço direito e as vezes braço esquerdo também, com quem fiz quase tudo relacionado a graduação junto, seja sendo ajudado, ajudando ou discutindo prestes a sair no soco porque as ideias não batiam; não sei se teria conseguido chegar aqui sem sua parceria, obrigado.

Por fim agradeço aos meus orientadores, professores Pedro e Arnaldo não só pelo apoio no desenvolvimento do TCC, mas por todo seu empenho ao longo da graduação. Agradeço igualmente aos demais professores que prezam pelo ensino e honram sua função de repassar o conhecimento que adquirem.

LISTA DE FIGURAS

LISTA DE SIGLAS

CBIR	Content-Based Image Retrieval
CIELAB	CIE L*a*b*
CIELUV	CIE L*u*v*
CMY	Cyan-Magenta-Yellow
CNNs	Convolutional Neural Networks
HLS	Hue-Luminosity-Saturation
HSB	Hue-Saturation-Brightness
HSV	Hue-Saturation-Value
IA	Inteligência Artificial
kNN	K-Nearest Neighbors
PCA	Principal Component Analysis
RGB	Red-Green-Blue
RNAs	Redes Neurais Artificiais
SVM	Support Vector Machine
TBIR	Text-based Image Retrieval
YIQ	Luminance, In-phase, Quadrature

SUMÁRIO

1 INTRODUÇÃO

É cada vez mais notável a expansão e produção de dados visuais por parte de empresas, organizações ou indivíduos, bem como o crescimento da popularidade de mídias sociais como o Facebook[1] e o Youtube[2] que são meios de difusão e compartilhamento de imagens e vídeos (SAHBI et al., 2013). Junto a isso surge uma necessidade com relação ao estudo e desenvolvimento de técnicas eficientes e com menor custo computacional para a detecção e classificação de imagens de forma automatizada, podendo assim ser aplicado para diversos fins. Pode-se tomar como exemplo no qual durante o processo de criação de um logotipo, há a possibilidade de se realizar uma busca comparativa em uma base de dados contendo logotipos existentes, afim de determinar se já não existe algo semelhante, ou ainda buscar por situações de uso não autorizado de uma determinada marca.

O reconhecimento de padrões está presente no dia à dia de todos. O simples ato de olhar para um objeto qualquer e o identificar através de algumas de suas características, como forma, tamanho ou cor, ou ainda, reconhecer um rosto familiar, envolve o processo de reconhecimento de padrões. O reconhecimento de padrões é uma área da ciência cujo o objetivo é classificar objetos em um determinado número de classes ou categorias, sendo que esses objetos podem ser imagens, sinais ou qualquer medida que precise ser classificada (THEODORIDIS; KOUTROUMBAS, 2009).

O processo de identificar e classificar um determinado objeto, logotipo, rosto de uma pessoa ou mesmo uma cadeia de caracteres presente em uma imagem pode parecer simples, tendo em vista que para o cérebro humano trata-se de uma tarefa corriqueira, isto porque o ser humano possui a capacidade única de reconhecer padrões, quando enxerga algo faz a coleta de informações para identificar, isolar, associar e reconhecer formas, sons ou conceitos (HELFER et al., 2006). Entretanto, do ponto de vista computacional, há uma dificuldade muito maior para se alcançar esse tipo de resultado, visto que mesmo os sistemas mais modernos não são capazes de processar de forma eficiente a quantidade de informações necessária para se atingir a mesma capacidade do cérebro humano (HERNANDEZ, 2011).

Para que um computador seja capaz de executar a tarefa de extrair características de

[1] https://www.facebook.com
[2] https://www.youtube.com

uma imagem e posteriormente realizar a classificação dos objetos de acordo com as classes correspondentes, ele deve ser treinado para tal, e no caso de imagens é comum que as mesmas sejam previamente submetidas à técnicas de processamento de imagens. O processamento de imagens serve para realizar melhorias na imagem, com o objetivo de realçar características específicas (GONZALEZ; WOODS, 2000). Um grande facilitador ao se trabalhar com o processamento de imagens é a possibilidade de se fazer uso da biblioteca OpenCV, que possui diversas funções implementadas com o objetivo de tornar mais rápido e eficiente o trabalho na área de visão computacional, tendo aplicações em áreas como linhas de produção, na área médica, segurança, robótica, dentre outras (BRADSKI; KAEHLER, 2008).

O processo de classificação dos objetos por parte do computador é possível através da área de Inteligência Artificial, mais precisamente do aprendizado de máquina, que tem como finalidade construir sistemas inteligentes a partir de dados. O processo de aprendizado é também conhecido como treinamento, e após o mesmo, o sistema pode ser utilizado para realizar classificações ou estimar saídas. Ainda pode-se definir o aprendizado de máquina como o campo de estudos que fornece ao computador a capacidade de aprender sem ser explicitamente programado (SIMON, 2013). O aprendizado de máquina pode ser representado através de métodos tradicionais, como regressão e *Support Vector Machines* (SVMs) ou então através de abordagens consideradas recentes como o *deep learning* (BROWN, 2015). No desenvolvimento desse trabalho, será feito o uso de *deep learning*, por ser uma área em desenvolvimento e que vem apresentando ótimos resultados, sendo utilizado por grandes corporações ao redor do mundo para diversos fins, como detecção de fraudes, análise de perfil de clientes, diagnósticos médicos, reconhecimento de fala, dentre outros (MURKANE, 2016).

1.1 OBJETIVO GERAL

Este trabalho tem como objetivo desenvolver um protótipo computacional capaz de realizar a classificação de logotipos de acordo com sua respectiva marca através de um sistema de Recuperação de Imagem Baseada em Conteúdo (CBIR - *Content-Based Image Retrieval*), fazendo uso de características extraídas por meio de *deep learning*.

1.2 OBJETIVOS ESPECÍFICOS

O objetivo principal pode ser dividido nos seguintes objetivos específicos:

- Realizar o treinamento de uma CNN;
- Modificar a estrutura da CNN treinada, afim de se obter vetores de características ao invés de predições;
- Desenvolver um sistema CBIR fazendo uso das características obtidas da CNN modificada;
- Realizar os testes com o sistema desenvolvido.

1.3 JUSTIFICATIVA

Notou-se que existe uma grande necessidade com relação ao processo de classificação de imagens de forma automatizada. Esse tipo de serviço, se realizado de forma manual torna-se repetitivo, cansativo e suscetível a falhas, principalmente quando se trabalha com uma grande quantidade de dados. Os meios para se alcançar o resultado esperado desse trabalho são objeto de estudo e interesse constante por parte de profissionais e pesquisadores na área da ciência da computação.

Nesse caso específico, o trabalho será aplicado tomando como base rótulos de produtos diversos, os quais devem ser classificados de acordo com sua marca. Porém, o mesmo processo é válido para outros casos, como a marca de uma empresa qualquer em imagens; princípios semelhantes se aplicam também para detecção de pessoas ou quaisquer objetos que venham a ser convenientes realizar a detecção e ou a classificação.

O objeto de estudo desse trabalho pode ser aplicado principalmente por empresas ou indivíduos, dos mais diversos segmentos e para os diversos fins, em campanhas de marketing, controle de qualidade de produtos, ou qualquer caso do qual possa se obter um resultado que tenha como meio o processo de classificação de imagens. O sistema, após preparado para o caso específico em que será aplicado, irá tornar esse processo de classificação automatizado.

2 RECUPERAÇÃO DE IMAGEM BASEADA EM CONTEÚDO

A recuperação de imagem baseada em conteúdo (*Content-based image retrieval - CBIR*) tem como objetivo realizar a busca por imagens utilizando como critério seu conteúdo visual (SUGAMYA et al., 2016). Logo, a partir de uma imagem fornecida pelo usuário, o sistema deve ser capaz de buscar por imagens similares dentro da base de dados. O conteúdo visual, é por sua vez constituído por atributos como cor, textura e forma, que são extraídos de cada imagem pertencente a base de dados e armazenados no que é chamado vetor de características (LUX, 2011), dessa maneira cada imagem na base de dados passa a ter seus atributos representados de forma numérica através desses vetores (SUGAMYA et al., 2016). Geralmente em um sistema CBIR, a similaridade entre as imagens é realizada utilizando funções de distância entre pontos nos vetores que representam cada uma das imagens, como a distância Euclidiana ou a similaridade do cosseno (ZHANG et al., 2002).

Basicamente existem duas formas de recuperação de imagem, baseada em conteúdo e textual (TBIR - *Text-based Image Retrieval*). Embora seja possível a organização através de tópicos para facilitar a navegação, o fato da grande maioria dos sistemas TBIR necessitarem que a notação seja realizada manualmente torna a tarefa exaustiva e trabalhosa ao se tratar de uma grande quantidade de dados, tornando o processo pouco escalável. A partir da década de 1990, o processo de recuperação baseada em texto tornou-se problemático por conta do grande aumento no número de imagens produzidas por dispositivos e aplicações comerciais. Na tentativa de se contornar os problemas encontrados em sistemas baseados em texto, surgiram os sistemas CBIR, que por sua vez já resolviam o problema da notação manual, que além de ser um processo pouco produtivo era subjetivo, tendo em vista que as notações feitas por uma determinada pessoa podem não ser compreensíveis à outras (GUAN et al., 2012).

De acordo com Shivhare (2015) qualquer sistema CBIR é composto por pelo menos quatro passos:

- Extração de características e indexação das imagens na base de dados, de acordo com as características que são relevantes, como cor, forma, textura ou mesmo qualquer combinação das mesmas;
- Extração de características da imagem usada na busca;

- Comparar a imagem de busca com as imagens da base de dados, buscando as que apresentam maior similaridade;
- Exibição dos resultados da busca ao usuário.

Na Figura 1 é ilustrada a arquitetura típica de um sistema CBIR.

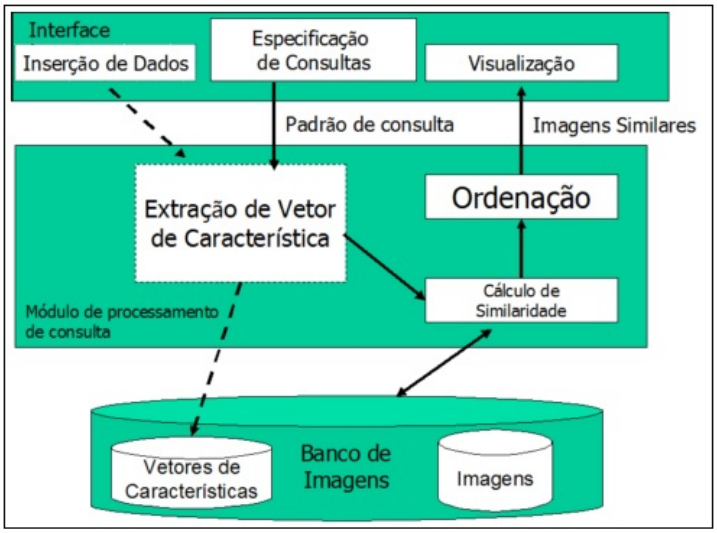


Figura 1 – Arquitetura típica de um sistema CBIR.
Fonte: Torres e Falcão (2008)

Ao se comparar um sistema CBIR com o Reconhecimento de Padrões, é possível dizer que no segundo caso, deve ser fornecido pelo usuário uma imagem a ser testada, essa imagem por sua vez é classificada em um determinado grupo, logo todas as imagens contidas nesse mesmo grupo são retornadas como resultado da pesquisa (BISHOP, 2006). Segundo Torres e Falcão (2008) dois aspectos fundamentais diferenciam um sistema CBIR do Reconhecimento de Padrões. O primeiro é de que o usuário que realiza a busca em um sistema CBIR é o responsável por julgar se as informações associadas às imagens retornadas são relevantes ou não para a consulta. Disso surge um problema, tendo em vista que o conceito de relevância do usuário pode não coincidir com o agrupamento realizado pelo sistema, esse problema é denominado como gap-semântico (*semantic gap*) (SHIVHARE, 2015). O segundo ponto é que muitas vezes, para uma mesma consulta, diferentes usuários podem julgar a relevância das imagens retornadas de maneira distinta, portanto, tentativas de classificação ou agrupamentos das imagens na base de dados que visam uma melhora na eficiência da consulta podem ajudar muito ou pouco dependendo do usuário e sua interpretação, porém o problema não é resolvido por completo.

2.1 FUNDAMENTOS DE PROCESSAMENTO DIGITAL DE IMAGENS

De acordo com Gonzalez e Woods (2000) o processamento de imagens digitais tem como objetivo a melhoria da informação visual para a interpretação humana e o processamento de dados em cenas para a percepção automática através de máquinas.

O processamento de imagens leva em consideração a manipulação de imagens após serem capturadas por um determinado dispositivo, o qual pode ser uma câmera digital, *scanner*, tomógrafo, sensor infravermelho, sensor de ultra-som, satélite, radar, dentre outros (CONCI et al., 2008). Pode-se dizer que no processamento de imagens, são abrangidas operações que são realizadas sobre imagens e têm como resultado final outras imagens (SCURI, 2002).

2.1.1 Etapas de um Sistema de Processamento de Imagens

Um sistema de processamento de imagens pode ser decomposto em diversas etapas. Para Pedrini e Schwartz (2008) e Gonzalez e Woods (2000) o processo é dividido em cinco etapas principais que são ilustradas na Figura 2, essas etapas por sua vez são capazes de produzir um resultado a partir do domínio do problema.

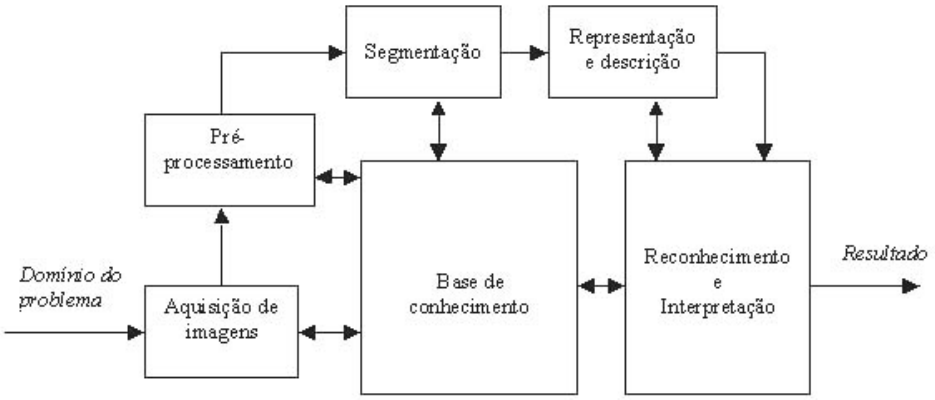


Figura 2 – Etapas de um sistema de processamento de imagens.
Fonte: Gonzalez e Woods (2000)

A seguir é apresentada uma breve descrição sobre cada uma das etapas que compõem

um sistema de processamento de imagens.

- **Aquisição:** a imagem é capturada através de um dispositivo imageador qualquer ou sensor. Alguns aspectos importantes a se ressaltar sobre essa etapa são as condições de iluminação da cena, resolução e o número de níveis de cinza ou cores da imagem digitalizada (PEDRINI; SCHWARTZ, 2008). Segundo Conci et al. (2008) imagens com poucos detalhes podem ser digitalizadas em poucos tons e com baixa resolução, de forma análoga, imagens com grande riqueza de detalhes devem ser digitalizadas com alta resolução e uma quantidade de níveis tonais elevada. Na Figura 3 é possível visualizar como resoluções menores apresentam uma grande redução na qualidade da imagem, na primeira a resolução (256 x 160) apresenta uma boa definição, enquanto que ao passo em que a resolução diminui nas imagens subsequentes (128 x 80) e (64 x 40), a imagem perde nitidez;

Figura 3 – Efeito da redução da resolução na qualidade da imagem.
Fonte: Conci et al. (2008)

- **Pré-processamento:** as imagens adquiridas no processo de aquisição podem apresentar imperfeições decorrentes de condições de iluminação ou características específicas dos dispositivos (PEDRINI; SCHWARTZ, 2008). Para Gonzalez e Woods (2000), a etapa de pré-processamento tem como objetivo destacar detalhes da imagem que são importantes para as etapas seguintes. É comum que nessa etapa sejam utilizadas técnicas para realçar o contraste, remover ruído, corrigir o foco. Dependendo do caso, a formulação matemática envolvida pode ser extremamente complexa, tornando o custo computacional muito alto (CONCI et al., 2008);
- **Segmentação:** de acordo com Pedrini e Schwartz (2008), a etapa de segmentação é responsável por realizar a extração e identificação de áreas de interesse contidas na imagem, baseando-se principalmente na detecção de descontinuidades (bordas) ou similaridades (regiões) da imagem. De forma simplificada, pode se considerar que o objetivo da segmentação é separar o conteúdo da imagem entre fundo e objeto;
- **Representação e descrição:** essa etapa também pode ser chamada de extração de atributos ou características (CONCI et al., 2008). Na parte de representação, é necessário

que sejam utilizadas estruturas adequadas para se armazenar e manipular os objetos de interesse extraídos da imagem. No processo de descrição é que são extraídas as características ou propriedades que podem vir a ser úteis na discriminação entre classes de objetos. As características são normalmente descritas por atributos numéricos que formam um vetor de características (PEDRINI; SCHWARTZ, 2008);

- **Reconhecimento e interpretação:** tanto para Gonzalez e Woods (2000) quanto para Pedrini e Schwartz (2008), o processo de reconhecimento é responsável por atribuir um identificador ou rótulo aos objetos da imagem. Ao mesmo passo que a interpretação é responsável por atribuir um significado a um conjunto de entidades rotuladas.

Toda a forma de conhecimento sobre o domínio do problema está codificada em um sistema de processamento de imagens na forma de uma base de conhecimento. Essa base de conhecimento é dependente da aplicação, sendo que seu tamanho e complexidade podem variar de acordo com o problema proposto. A base de conhecimento deve ser utilizada para guiar a comunicação entre os módulos de processamento, a fim de executar uma determinada tarefa (PEDRINI; SCHWARTZ, 2008).

2.2 EXTRAÇÃO DE CARACTERÍSTICAS

A extração de características é uma das principais etapas em um sistema CBIR. Segundo (LUX, 2011), uma imagem no seu estado original, não pode ser diretamente utilizada na maioria das tarefas de visão computacional. Duas razões principais estão por trás disso, a grande dimensionalidade da imagem torna muito difícil utilizá-la como um todo. O segundo ponto é que muitas das informações contidas na imagem são consideradas redundantes. Portanto é conveniente se fazer uso apenas de uma representação das informações mais significantes. Este processo de buscar e extrair as informações mais significantes é denominado extração de características, e tem como resultado um vetor de características (BUGATTI et al., 2014).

A pesquisa por trás do problema da extração de características é principalmente baseada em técnicas fundamentais da área de visão computacional e processamento de imagens. O conteúdo visual de uma imagem pode ser muito geral (cor, textura, forma) ou muito específico (rostos de pessoas, impressões digitais). No caso do domínio específico, a extração depende da aplicação e de um conhecimento aprofundado do domínio especificado (GUAN et al., 2012).

Serão apresentadas a seguir algumas das principais técnicas para extração de características de domínio geral.

2.2.1 Cor

De acordo com Shengjiu (2001), a cor é um dos elementos mais importantes para tornar possível o reconhecimento de imagens por parte da percepção humana, tendo em vista que ela é utilizada diariamente para fazer distinção de objetos, lugares e até mesmo da hora do dia. Características extraídas das cores podem fornecer informações extremamente úteis à recuperação de imagens. Segundo Guan et al. (2012), cor é a característica visual mais utilizada em sistemas CBIR pelo fato de ser relativamente robusta e independente do tamanho e orientação da imagem.

As características de cor são extraídas para um determinado espaço de cor. Os espaços de cor podem ser divididos entre orientados a hardware ou à percepção humana. Dentre os espaços de cor orientados a hardware estão o RGB (*Red-Green-Blue*), CMY (*Cyan-Magenta-Yellow*) e YIQ (*Luminance, In-phase, Quadrature*). Já nos espaços orientados à percepção humana pode-se citar o HSV (*Hue-Saturation-Value*), HLS (*Hue-Luminosity-Saturation*), HSB (*Hue-Saturation-Brightness*), CIELAB (*CIE L*a*b**) e o CIELUV (*CIE L*u*v**) (GONZALEZ; WOODS, 2000).

Em um sistema CBIR, realizar a escolha do espaço de cor apropriado é de extrema importância para se obter resultados mais precisos. A seguir são descritos alguns dos principais espaços de cor citados anteriormente:

- **RGB (*Red-Green-Blue*)**: este espaço de cor baseia-se nas três cores primárias (vermelho, verde e azul) para a formação das demais cores. O RGB pode ser representado por um vetor de três coordenadas variando de zero a um. Quando todas estão definidas em 0 se obtém o preto, da mesma forma que ao se estipular 1 a todas as coordenadas se obtém branco como resultado (SHENGJIU, 2001). Sua representação pode se dar também através da Figura 4 a qual ilustra o cubo RGB, que possui vértices com as cores principais (vermelho, verde e azul), as secundárias (ciano, magenta e amarelo) além do branco e preto (KHOKHER; GOBINDGARH, 2008);
- **HSV (*Hue-Saturation-Value*)**: neste modelo, as cores são representadas por três componentes: matiz, saturação e valor. A matiz corresponde ao tipo de cor em si (azul

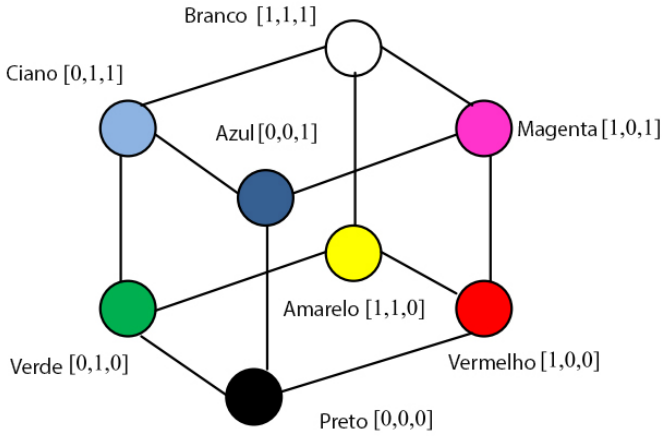


Figura 4 – Modelo de cor RGB.
Fonte: Khokher e Gobindgarh (2008) (adaptado)

ou verde por exemplo), a saturação determina o quão vibrante a cor é, enquanto que o valor corresponde ao brilho da cor (PEDRINI; SCHWARTZ, 2008). O HSV pode ser representado graficamente por uma pirâmide hexagonal ilustrada na Figura 5. Segundo Yu et al. (2009) e Khokher e Gobindgarh (2008), esse é um dos modelos que se destaca no processamento de imagens, e é amplamente utilizado na área da computação gráfica. Por esse fato, é conveniente muitas vezes realizar a transformação de um espaço de cor RGB para o formato HSV, o que pode ser feito através das Equações 1, 2 e 3, onde $M = max(R,G,B)$ e $m = min(R,G,B)$ (PEDRINI; SCHWARTZ, 2008);

$$H = \begin{cases} 60\dfrac{(G-B)}{(M-m)} & \text{, se } M = R, \\ 60\dfrac{(B-R)}{(M-m)} + 120 & \text{, se } M = G, \\ 60\dfrac{(R-G)}{(M-m)} + 240 & \text{, se } M = B. \end{cases} \tag{1}$$

$$S = \begin{cases} \dfrac{(M-m)}{(M)} & \text{, se } M \neq 0, \\ 0 & \text{, caso contrário .} \end{cases} \tag{2}$$

$$V = M \tag{3}$$

- **CIELAB e CIELUV (*CIE L*a*b* e CIE L*u*v**)**: Tratam-se de modelos de cor uniformes e são totalmente independentes do dispositivo utilizado para sua exibição. No primeiro modelo o L^* representa a luminosidade, a^* e b^* representam a distância entre as cores magenta e verde, amarelo e azul, respectivamente. Ambos os modelos são

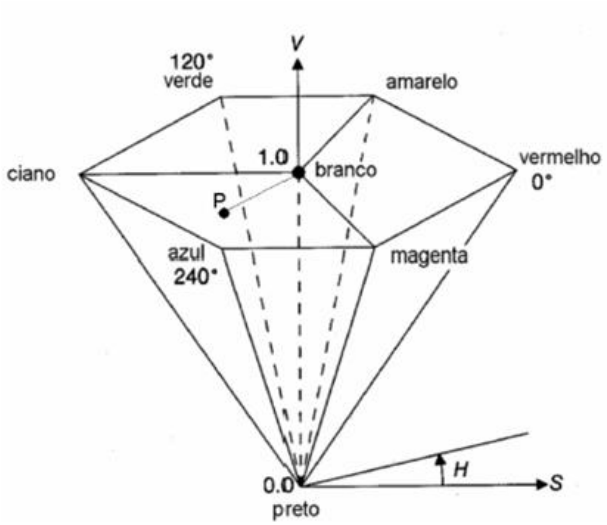


Figura 5 – Modelo de cor HSV.
Fonte: Pedrini e Schwartz (2008)

apresentados nas Figuras 6 e 7.

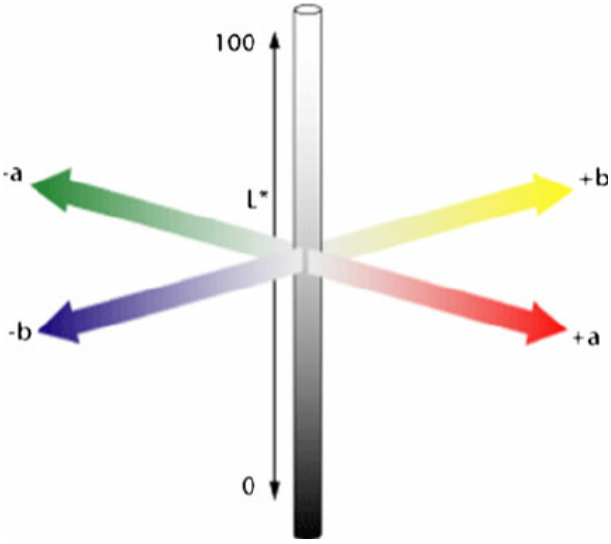


Figura 6 – Modelo CIELAB
Fonte: Schils (2010)

Para o processo de extração de características, as cores podem ser representadas de diferentes maneiras, como por exemplo histogramas, momentos de cor, correlogramas, dentre outros.

- **Histograma:** Pedrini e Schwartz (2008) definem um histograma de uma imagem como a distribuição dos níveis de cinza da imagem, o qual pode ser representado por um gráfico indicando o número de pixels na imagem para cada nível de cinza. Trata-se do principal método para a representação de informações de cores em um sistema CBIR (SWAIN;

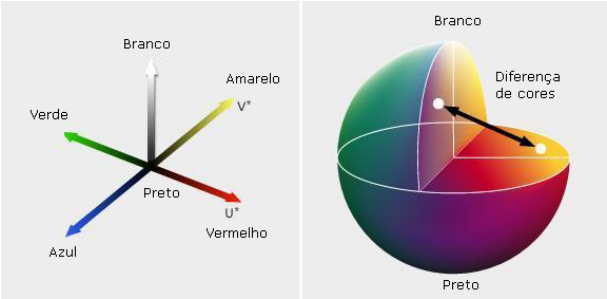


Figura 7 – Modelo CIELUV
Fonte: Schils (2010)

BALLARD, 1991). Uma imagem possui um único histograma, porém a recíproca não é verdadeira, tendo em vista que um histograma não possui informação espacial, apenas valores de intensidade. A Figura 8 ilustra um caso onde duas imagens diferentes apresentam o mesmo histograma.

Normalmente para se fazer a comparação entre histogramas de imagens são utilizadas medidas de distância como a Euclidiana (Equação 4) ou Manhattan (Equação 5):

$$L(P,Q) = \sqrt{\sum_{i=1}^{n} (P_i - Q_i)^2} \tag{4}$$

$$L(P,Q) = \sum_{i=1}^{n} |P_i - Q_i| \tag{5}$$

O fato do histograma não apresentar informações espaciais da imagem pode ser visto como um ponto negativo, considerando que em grandes bases de dados é possível encontrar imagens que não apresentam similaridade mas que possuem um histograma parecido, e em alguns casos até mesmo exatamente igual;

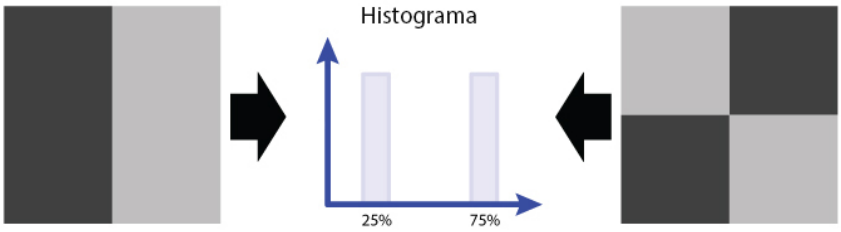


Figura 8 – Exemplo de diferentes imagens que apresentam o mesmo histograma.
Fonte: Kunzman (2016)

- **Momentos de cor:** A utilização de momentos de cor como um método de extração

de características foi proposto pela primeira vez por Stricker e Orengo (1995) com o intuito de analisar a distribuição de cores em imagens e comparar a similaridade entre elas. Segundo Guan et al. (2012), é provado que há a possibilidade de se interpretar a distribuição de cor em uma imagem como um distribuição de probabilidade, especialmente os momentos de primeira ordem (média), segunda ordem (variância) e terceira ordem (assimetria). Esses três momentos podem ser definidos matematicamente pelas Equações 6, 7 e 8, onde f_{ij} é o i-ésimo componente de cor do pixel j, e N é a quantidade de pixels da imagem. De acordo com Guan et al. (2012), os modelos de cor CIELAB e CIELUV são melhores com relação ao HSV ao se utilizar dessa técnica. Khokher e Gobindgarh (2008) afirmam que o método de extração por momentos de cor também possui o problema de não guardar informações espaciais da imagem, assim como ocorre com o histograma;

$$\mu_i = \frac{1}{N}\sum_{j=1}^{N} f_{ij} \tag{6}$$

$$\sigma_i = \left(\frac{1}{N}\sum_{j=1}^{N}\left(f_{ij}-\mu_i\right)^2\right)^{\frac{1}{2}} \tag{7}$$

$$s_i = \left(\frac{1}{N}\sum_{j=1}^{N}\left(f_{ij}-\mu_i\right)^3\right)^{\frac{1}{3}} \tag{8}$$

- **Correlograma:** O correlograma de cor foi proposto para caracterizar a distribuição de cores dos pixels de uma imagem, porém diferentemente dos métodos de histograma e momentos de imagem, o correlograma incorpora aos seus dados informações espaciais da imagem, o que evita diversos problemas que estão presentes nos outros métodos (KHOKHER; GOBINDGARH, 2008). Ainda de acordo com Ponti Jr. (2011), um correlograma descreve a distribuição global da correlação entre a localização espacial de cores.

Os descritores apresentados são alguns dos mais populares, sendo que ainda existem diversas variações dos mesmos. Os descritores de cor estão diretamente relacionados com operações matemáticas aplicadas aos valores dos pixels dentro de uma determinado espaço de cor (KHOKHER; GOBINDGARH, 2008).

2.2.2 Textura

Segundo Conci et al. (2008), embora possa-se dizer que textura é um termo intuitivo e de grande importância na área da visão computacional, não existe uma definição precisa. De maneira geral, o termo refere-se a um determinado padrão visual que possui algumas propriedades de homogeneidade que não resultam simplesmente de uma cor ou intensidade, de forma que pode ser definida como o aspecto visual de uma superfície. Já para Gonzalez e Woods (2000) a textura é descrita por medidas que quantificam suas propriedades de suavidade, rugosidade e regularidade. As texturas são propriedades naturais de todas as superfícies, desde nuvens, paredes, grama, tecidos, entre outros.

A textura é uma das características mais importantes utilizadas na identificação de objetos ou regiões de interesse em uma imagem (GUAN et al., 2012). Ainda segundo o autor, os métodos de representação de texturas podem ser classificados em duas categorias, sendo elas estatística e estrutural. Já para Khokher e Gobindgarh (2008) e Gonzalez e Woods (2000), além dessas duas classificações, pode-se definir também uma terceira denominada espectral.

De acordo com Guan et al. (2012), no passado os estudos sobre extração de características através das texturas se baseavam-se basicamente na abordagem estrutural. Somente a partir de 1970 foram adotados em maior escala os métodos estatísticos. Em 1973 foi proposto por Haralick et al. (1973) um modelo fazendo utilização de uma matriz de co-ocorrência para a representação das características das texturas.

2.2.2.1 Abordagem Estatística

Na abordagem estatística as texturas são caracterizadas utilizando propriedades estatísticas dos níveis de cinza dos pixels que compõem uma imagem. Nas imagens, é comum que haja a ocorrência periódica de alguns níveis de cinza, logo é calculada a distribuição espacial desses níveis. A ideia principal por trás dessa abordagem é representar a textura através de propriedades estatísticas que definem a forma de distribuição e o relacionamento entre os níveis de cinza. Os métodos que utilizam a abordagem estatística não buscam compreender explicitamente a estrutura hierárquica da textura, mas tentam representar a textura indiretamente por propriedades não-determinísticas que definem distribuições e relacionamentos entre os níveis de cinza dos pixels pertencentes a uma imagem (PEDRINI; SCHWARTZ, 2008).

- **Matrizes de Co-Ocorrência:** Como já citado anteriormente, esse método foi proposto por Haralick et al. (1973), desde então é um dos métodos mais utilizados, mesmo com diversas variações com relação à forma com que são calculadas as matrizes. A ideia principal é que cada elemento da matriz de co-ocorrência represente a frequência com que um pixel com nível de cinza i e outro com nível de cinza j ocorrem na imagem, separados de uma distância d, na direção θ ou separados entre si de Δx colunas e Δy linhas. O número de linhas e colunas dessa matriz é proporcional à quantidade de níveis de cinza contidos na textura, ou seja, independe das dimensões da textura, acarretando, dessa maneira, perda do relacionamento espacial nela contida.

  Para realizar a análise das propriedades contidas nas texturas, Haralick et al. (1973) propôs algumas medidas estatísticas, também chamadas de descritores, a serem calculadas a partir das matrizes de co-ocorrência. Segundo Baraldi e Parmiggiani (1995), seis desses descritores são considerados relevantes, sendo eles: *energia, entropia, contraste, variância, correlação e homogeneidade*.

  - **Energia** (Equação 9): Retorna a soma dos elementos elevados ao quadrado dentro da matriz de co-ocorrência de tons de cinza. A faixa de valores varia entre 0 e 1, sendo que a energia possui valor 1 para uma imagem constante (que possui o mesmo tom de cinza por toda sua extensão).

$$\sum_{i=0}^{N_g-1}\sum_{j=0}^{N_g-1} g^2(i,j) \tag{9}$$

  - **Entropia** (Equação 10): Mede a informação contida em g; muitos valores nulos representam pouca informação.

$$-\sum_{i=0}^{N_g-1}\sum_{j=0}^{N_g-1} g(i,j)\cdot\log\left(g(i,j)\right) \tag{10}$$

  - **Contraste** (Equação 11) ou **Variância** (Equações 12, 13): Retorna uma medida do contraste entre as intensidades de um pixel analisado e do pixel vizinho. Para uma imagem constante o resultado será 0.

$$\sum_{i=0}^{N_g-1}\sum_{j=0}^{N_g-1} (i-j)^2\cdot g(i,j) \tag{11}$$

$$\sum_{i=0}^{N_g-1}\sum_{j=0}^{N_g-1} (i-\mu_i)^2\cdot g(i,j) \tag{12}$$

$$\sum_{i=0}^{N_g-1}\sum_{j=0}^{N_g-1}\left(j-\mu_j\right)^2\cdot g\left(i,j\right) \tag{13}$$

– **Correlação** (Equação 14): Retorna uma medida de quão correlacionado está um pixel com o seu vizinho. A comparação é realizada em todos os pixels da imagem. Os valores possíveis variam de -1 a 1. A correlação é 1 em uma imagem totalmente correlacionada ou -1 para uma completamente descorrelacionada.

$$\sum_{i=0}^{N_g-1}\sum_{j=0}^{N_g-1} g\left(i,j\right)\cdot\frac{(i-\mu)\cdot(j-\mu)}{\sigma^2} \tag{14}$$

– **Homogeneidade** (Equação 15): Retorna um valor que representa a proximidade da distribuição dos elementos em relação a diagonal da matriz. Varia entre 0 e 1, sendo que 1 representa uma matriz diagonal.

$$\sum_{i=0}^{N_g-1}\sum_{j=0}^{N_g-1}\frac{g\left(i,j\right)}{(1+|i-j|)} \tag{15}$$

- **Medidas baseadas na Distribuição de Níveis de Cinza:** Para Pedrini e Schwartz (2008), considerar apenas a intensidade de cada pixel de maneira isolada pode apresentar algumas desvantagens, porém o custo computacional requerido para a extração de medidas é baixo, o que é uma característica muito importante para alguns sistemas.
  Dentre essas medidas estão a média (Equação 16) e a variância (Equação 17), que representam o valor esperado na distribuição dos níveis de cinza presentes na textura e descreve quanto os valores estão dispersos em torno da média, respectivamente. Nas equações, g_i representa o tom de cinza para o i-ésimo pixel e n o número de pixels presentes na textura (PEDRINI; SCHWARTZ, 2008).

$$\mu=\frac{1}{n}\sum_{i=1}^{n} g_i \tag{16}$$

$$\sigma^2=\frac{1}{n}\sum_{i=1}^{n}\left(g_i-\mu\right)^2 \tag{17}$$

  O grau de assimetria da distribuição (Equação 18) é um indicador da concentração de valores em relação à mediana. A curtose (Equação 19) indica o achatamento da função de distribuição e apresenta valores negativos em distribuições com forma mais achatada que a Gaussiana, que por sua vez apresenta assimetria nula.

$$s=\frac{1}{n\sigma^3}\sum_{i=1}^{n}\left(g_i-\mu\right)^3 \tag{18}$$

$$k = \left(\frac{1}{n\sigma^4} \sum_{i=1}^{n} (g_i - \mu)^4 \right) - 3 \tag{19}$$

Pode-se também fazer uso do histograma dos níveis de cinza para a extração de medidas estatísticas. De acordo com Pedrini e Schwartz (2008) ele fornece informações baseadas na consideração individual dos níveis de cinza contidos na textura. A Equação 20 ilustra como a função de massa de probabilidade pode ser determinada, onde $h(i)$ denota o número de ocorrências de pixels apresentando intensidade i.

$$P(i) = \frac{h(i)}{n} \tag{20}$$

A partir do histograma podem ser calculadas duas medidas, a energia (Equação 21) e a entropia (Equação 22). Nessas equações, H_g representa o tom de cinza máximo.

$$E = \sum_{i=0}^{H_g} (P(i))^2 \tag{21}$$

$$H = -\sum_{i=0}^{H_g} P(i)\, log\, (P(i)) \tag{22}$$

2.2.2.2 Abordagem Estrutural

A ideia básica é que uma "primitiva de textura" simples pode ser utilizada para a formação de padrões complexos de textura através de um conjunto de regras que limitam o número de arranjos possíveis das primitivas (GONZALEZ; WOODS, 2000). Segundo Pedrini e Schwartz (2008), após a identificação das primitivas que a compõem, são utilizadas duas classes para a extração de características. A primeira utiliza medidas extraídas das primitivas para descrever a textura, enquanto a segunda extrai regras para descrever a disposição espacial e o relacionamento existente entre as primitivas.

A vantagem de se fazer uso dos métodos estruturais está no fato que eles apresentam uma boa descrição simbólica da textura, característica muito útil para a síntese de texturas. Em contrapartida, na análise de texturas, os métodos estruturais não apresentam boa adaptação pelo fato de apenas atuarem de maneira satisfatória em texturas regulares, o que praticamente nunca ocorre em imagens naturais (PEDRINI; SCHWARTZ, 2008).

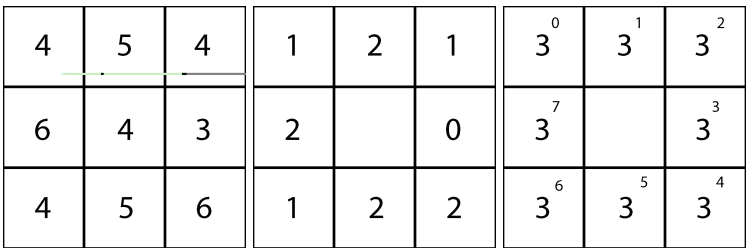
- **Unidade de Textura:** Wang e He (1990) propõem um conceito chamado unidade de textura, baseando-se na ideia de que uma imagem texturizada pode ser considerada como um conjunto de pequenas unidades essenciais. Enquanto tais unidades caracterizam a informação local de um dado pixel em relação aos seus vizinhos, medidas extraídas a partir de todas as unidades contidas na imagem revelam o aspecto global da textura (PEDRINI; SCHWARTZ, 2008).

  Partindo de um exemplo onde se tem um grupo de 3x3 pixels, composto pelos elementos $\{g_0, g_1, g_2 ... g_8\}$, onde g_0 representa o tom de cinza do pixel central e os demais denotam os tons de cinza de seus vizinhos mais próximos. Wang e He (1990) definem como unidade de textura o conjunto $TU = \{e_1, e_2 ... e_8\}$ em que cada e_i é determinado por meio da Equação 23.

$$e_i = \begin{cases} 0 & \text{, se } g_i < g_0, \\ 1 & \text{, se } g_i = g_0, \\ 2 & \text{, se } g_i > g_0. \end{cases} \tag{23}$$

  Baseando-se nas possíveis configurações para cada unidade de textura, cria-se uma assinatura conforme define a Equação 24.

$$N_{TU} = \sum_{i=1}^{8} 3^{(i-1)} e_i \tag{24}$$



(a) (b) (c)

Figura 9 – Exemplo da unidade de textura.
Fonte: Pedrini e Schwartz (2008)

Analisando a Figura 9, faz-se uso da Equação 23, cada valor de e_i é calculado e apresentado na Figura (b). De acordo com a ordenação estabelecida em (c), é utilizado a Equação 24 para se obter o número da unidade de textura.

2.2.2.3 Abordagem Espectral

De acordo com Pedrini e Schwartz (2008) os métodos de análise de texturas baseados em processamento de sinais possuem a característica de extrair descritores a partir da representação obtida após realizada a aplicação de transformações na imagem de entrada. Gonzalez e Woods (2000) sugerem que o espectro de Fourier pode ser adaptado para a descrição da orientação de padrões periódicos ou quase periódicos em uma imagem. Ainda conclui que, embora esses padrões de textura sejam facilmente distinguíveis como concentrações de agrupamentos de alta-energia no espectro, são em grande parte dos casos, difíceis de se detectar com métodos espaciais devido a natureza local dessas técnicas.

O espectro de Fourier possui três características que são úteis para a descrição de texturas:

- Picos proeminentes no espectro fornecem a direção dos padrões de textura;
- A posição dos picos no plano da frequência fornece o período espacial fundamental dos padrões;
- A eliminação de quaisquer componentes periódicos através de filtragem deixa os elementos não-periódicos na imagem, que podem ser descritos por técnicas estatísticas.

Após a transformada de Fourier, o espectro resultante quando efetuado o deslocamento do plano de frequências da origem, apresenta grande concentração de energia no centro no caso de imagens que possuem componentes de baixa frequência, enquanto que em imagens que possuem alta frequência espacial, essa energia fica mais dispersa, localizando-se inclusive em regiões muito distantes da origem (PEDRINI; SCHWARTZ, 2008).

É possível expressar o espectro em um sistema de coordenadas polares do formato $S(r,\theta)$, sendo S uma função resultante do espectro de Fourier, e r e θ variáveis comuns desse sistema, que representam um raio e um ângulo respectivamente. Uma descrição global é obtida por meio das Equações 25 e 26.

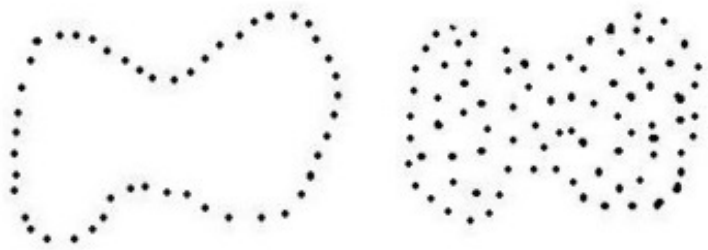
$$S(r) = \sum_{\theta=0}^{\pi} S_\theta(r) \tag{25}$$

$$S(\theta) = \sum_{r=1}^{R} S_r(\theta) \tag{26}$$

De acordo com Pedrini e Schwartz (2008) as funções unidimensionais $S(r)$ e $S(\theta)$, que constituem descrições de energia espectral da textura para uma imagem ou região, são utilizadas para a extração de características de texturas.

2.2.3 Forma

Segundo Khokher e Gobindgarh (2008), definir a forma de um determinado objeto é geralmente uma tarefa complicada. Embora a forma seja descrita verbalmente ou em figuras através de termos comuns ao entendimento (arredondado, elíptico, etc), o processamento de formas por parte de um sistema computacional não é tão simples, tendo em vista que há a necessidade de se descrever precisamente mesmo as formas mais complicadas. Uma boa representação de característica de forma para um determinado objeto deve ser independente com relação à translação, rotação e escala (GUAN et al., 2012). Os principais métodos para a descrição de formas podem ser categorizados entre baseados em fronteira ou baseados em região, da esquerda para a direita respectivamente, como ilustrado na Figura **??**.



(a) (b)

Figura 10 – Representação de formas baseada em fronteira (a) e baseada em região (b).
Fonte: (FACELI et al., 2011)

As formas de representação mais conhecidas para as duas categorias citadas são o descritor de Fourier e invariantes de momento. O primeiro faz uso da transformada de Fourier com base na fronteira como a característica de forma, enquanto que a invariante de momento utiliza os baseados em região, que por sua vez são invariáveis a transformações.

Ao se fazer uma comparação com os métodos de extração de características baseados em cor e textura, a forma é normalmente descrita após a segmentação da imagem em regiões ou objetos. Tendo em vista que um processo robusto e preciso de segmentação é difícil de se alcançar, a utilização de características baseadas em forma para a recuperação de imagens se limita à casos especiais onde os objetos ou regiões já estejam disponíveis.

Os descritores de forma geralmente são divididos entre descritores de borda (diâmetro, perímetro, curvatura, energia de deformação e descritor de Fourier) e descritores de região (área e circularidade).

3 APRENDIZADO DE MÁQUINA

Antes mesmo dos primeiros computadores programáveis serem concebidos, já era cogitado se um dia essas máquinas poderiam tornar-se inteligentes. Nos dias atuais a Inteligência Artificial (IA) é amplamente estudada e possui diversas aplicações práticas, como tarefas de rotina, reconhecimento de fala ou imagens, realizar diagnósticos médicos, dentre outros (GOODFELLOW et al., 2016).

Segundo Smola e Vishwanathan (2008), o aprendizado de máquina pode aparecer de diversas formas, porém um dos pontos mais importantes é buscar por se reduzir o número de problemas a serem solucionados a grupos mais genéricos, de uma forma onde não haja a necessidade de se elaborar uma solução completamente nova para cada aplicação. Da mesma forma, Domingos (2012) define que os algoritmos de aprendizado de máquina podem descobrir como realizar determinadas tarefas através da generalização de exemplos.

De acordo com Mitchell (2006), o campo do aprendizado de máquina busca responder a questão: "Como podemos construir sistemas computacionais que são capazes de evoluir de forma automatizada, e quais são as regras fundamentais que controlam todos os processos de aprendizado?" Segundo o autor, uma máquina "aprende" à respeito de um tarefa T, com performance P, e tipo de experiência E, se o sistema melhora sua performance P com relação à tarefa T seguindo sua experiência E.

O aprendizado de máquina pode ser classificado entre não supervisionado, supervisionado ou semi-supervisionado. O primeiro treina baseado em um conjunto de dados que contém muitas características, aprendendo propriedades úteis da estrutura desse conjunto de dados. No caso do aprendizado supervisionado, ocorre o treinamento, porém cada exemplo das características está associado a um rótulo ou destino (GOODFELLOW et al., 2016). Pode-se resumir que no aprendizado supervisionado a entrada e saída desejadas são fornecidas por um supervisor externo, que por sua vez indica se um comportamento é bom ou ruim para a rede. Já no aprendizado não supervisionado esse supervisor externo não existe, e apenas os padrões de entrada estão disponíveis para a rede (BRAGA et al., 2000). O aprendizado semi-supervisionado por sua vez é um meio termo entre os dois, tendo em vista que o conjunto de treinamento inclui algumas das saídas desejadas (BROWNLEE, 2015).

3.1 APLICAÇÕES

Uma das maneiras de se avaliar a importância do aprendizado de máquina é analisar a quantidade de aplicações do mundo real que fazem uso do mesmo, a seguir são apresentadas algumas áreas e casos de aplicação.

- **Visão Computacional:** Diversos sistemas de visão computacional fazem uso do aprendizado de máquina, como sistemas para reconhecimento de faces, classificação de imagens em geral. Um exemplo interessante é o serviço de correspondências dos Estados Unidos (US Post Office) que faz uso do aprendizado de máquina para identificar automaticamente endereços escritos à mão nas correspondências em 98% dos casos (Digital For All Now, 2015). A Figura 11 ilustra um exemplo de diferentes dígitos escritos à mão que podem ser utilizadas para treinamento;

Figura 11 – Exemplo de dígitos feitos à mão, utilizados para treinamento.
Fonte: Smola e Vishwanathan (2008)

- **Reconhecimento de fala:** O aprendizado de máquina é aplicado pelo simples fato de que a precisão é maior quando o sistema é treinado, em comparação com o mesmo sendo programado à mão (MITCHELL, 2006);
- **Controle de robôs:** É aplicado com sucesso em diversos casos envolvendo o controle de robôs ou veículos. É possível por exemplo fazer uso do aprendizado de máquina para elaborar estratégias de voo buscando estabilidade em helicópteros ou drones (WANG et al., 2012);
- **Ranqueamento de páginas web:** Quando se realiza uma pesquisa em um mecanismo de busca, são retornadas páginas relevantes à pesquisa, teoricamente em ordem baseada na relevância. Para que esse processo seja possível, o mecanismo de busca precisa saber o que é relevante perante a busca feita pelo usuário, e isso é possível através do aprendizado

de máquina (SMOLA; VISHWANATHAN, 2008);

- **Detecção de fraudes:** Nos últimos anos grandes empresas da área financeira passaram a fazer uso do aprendizado de máquina como ferramenta para identificar possíveis fraudes por parte dos usuários, levando em conta informações do histórico de movimentação do mesmo (KNORR, 2015).

Baseando-se nos exemplos citados é possível dizer que alguns métodos de aprendizado de máquina já são as melhores opções disponíveis em alguns casos particulares, principalmente em aplicações que possuem tarefas muito complexas para serem codificadas manualmente ou que a aplicação necessite adaptar-se de forma customizada ao seu ambiente. Por exemplo, é possível para um humano reconhecer uma determinada pessoa em uma fotografia, porém é inviável escrever um algoritmo capaz de realizar isso de forma genérica (MITCHELL, 2006).

3.2 REDES NEURAIS ARTIFICIAIS

De acordo com Braga et al. (2000) as Redes Neurais Artificias (RNAs) são uma forma de computação não algorítmica e que em algum nível relembram a maneira de funcionamento do cérebro humano.

Em uma abordagem convencional de programação, é dito ao computador o que, e como fazer, dividindo problemas em outros menores e definindo precisamente as tarefas necessárias para que o computador possa realizá-las facilmente. Por outro lado, em uma Rede Neural não é especificado ao computador como resolver o problema, ao invés disso, o mesmo "aprende" através da observação de dados, encontrando a solução por conta própria (NIELSEN, 2015).

Segundo Faceli et al. (2011) as RNAs são sistemas computacionais distribuídos, compostos de unidades de processamento simples e densamente interconectadas. Essas unidades são conhecidas como *neurônios artificiais* e têm a função de realizar a computação de funções matemáticas. As unidades são dispostas em uma ou mais camadas e interligadas por uma grande quantidade de conexões. Normalmente essas conexões possuem pesos associados, que ponderam a entrada recebida por cada neurônio da rede. Os pesos têm seus valores ajustados em um processo de aprendizado e codificam o conhecimento adquirido pela rede (BRAGA et al., 2000).

O neurônio é a unidade de processamento fundamental em uma RNA. As unidades de processamento desempenham um trabalho simples. Cada terminal de entrada do neurônio

recebe um valor, os valores recebidos então são ponderados e combinados por uma certa função matemática, logo a saída da função é a resposta do neurônio para a entrada (FACELI et al., 2011). Esse processo é ilustrado na Figura 12, onde w são os pesos e f_a representa a função responsável por gerar a saída.

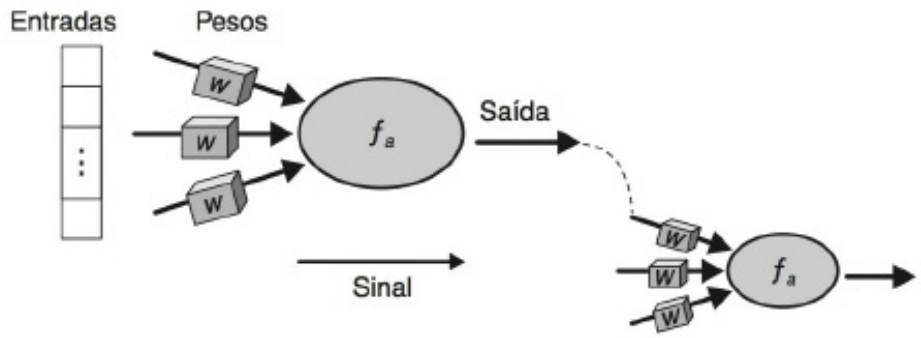


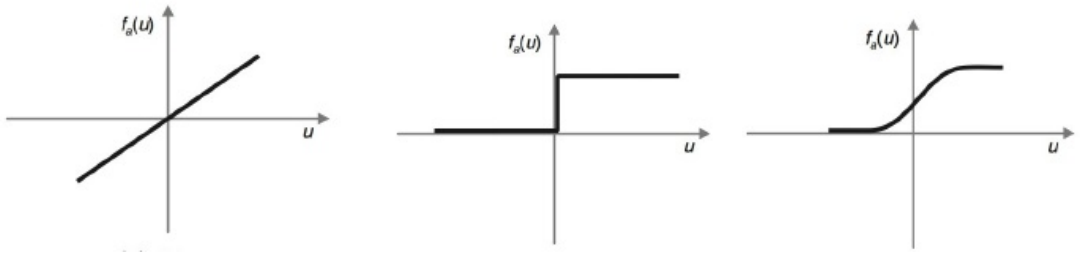
Figura 12 – Neurônio artificial.
Fonte: Faceli et al. (2011)

A saída de um neurônio é definida pela *função de ativação*. De acordo com Haykin (2001) as três formas mais comuns de função de ativação são a limiar, a linear e a sigmóide.

- **Função linear**: Pode ser representada na forma $y = \alpha x$, sendo α um número real que define a saída linear para os valores de entrada, y é a saída e x é a entrada (BRAGA et al., 2000);
- **Função de limiar**: Também referido como *Modelo de McCulloch-Pitts*, nesse modelo, a saída de um neurônio assume valor 1, se o campo local induzido daquele neurônio é não negativo e 0 caso contrário (HAYKIN, 2001), (FACELI et al., 2011);
- **Função sigmóide**: É a forma mais comum de função de ativação utilizada na construção de redes neurais artificias (HAYKIN, 2001). A função sigmóide representa uma aproximação contínua e diferenciável da função limiar (FACELI et al., 2011).

Os gráficos das três funções de ativação apresentadas são ilustrados na Figura 13.

Em uma RNA, a disposição dos neurônios pode ocorrer através de uma ou mais camadas. Quando mais de uma camada é utilizada, o neurônio pode receber em suas entradas valores de saída de neurônios da camada anterior, assim como também pode enviar seu valor de saída para terminais de entrada de neurônios da camada subsequente (FACELI et al., 2011). Uma rede que possui mais de uma camada é denominada rede multicamadas, sendo que além da camada de saída existente em uma RNA de camada única, há a presença de *camadas intermediárias ou camadas de neurônios ocultos* (HAYKIN, 2001). A Figura 14 ilustra uma RNA multicamadas composta pela camada de saída e duas camadas intermediárias.



(a) (b) (c)

Figura 13 – (a) Função Linear, (b) Função Limiar e (c) Função Sigmóide.
Fonte: (FACELI et al., 2011)

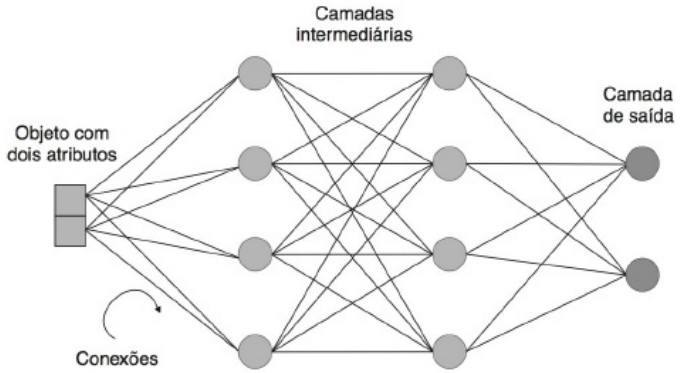


Figura 14 – Exemplo de RNA multicamadas.
Fonte: Faceli et al. (2011)

De acordo com Faceli et al. (2011), em uma rede multicamadas as conexões entre os neurônios podem apresentar diferentes padrões de conexão, sendo que de acordo com esses padrões, a rede pode ser classificada de três maneiras:

- **Completamente conectada:** ocorre quando os neurônios da rede estão conectados a todos os neurônios das camadas anterior e subsequente;
- **Parcialmente conectada:** ocorre quando os neurônios da rede estão conectados a apenas alguns neurônios das camadas anterior e subsequente;
- **Localmente conectada:** são redes parcialmente conectadas, em que os neurônios conectados a outros se encontram em uma região bem definida.

As redes podem ainda ser divididas entre *recorrentes* ou *feedforward*. Nas redes recorrentes, é possível que um neurônio receba em seus terminais de entrada a saída de um neurônio da mesma camada ou de uma camada posterior, sendo possível até mesmo receber sua própria saída em um de seus terminais de entrada. No caso das redes *feedforward* o neurônio recebe em suas entradas apenas a saída de neurônios das camadas anteriores, sendo que essa é

a implementação mais utilizada em aplicações práticas (BRAGA et al., 2000), (FACELI et al., 2011) e (HAYKIN, 2001).

3.3 DEEP LEARNING

A ideia por trás do *deep learning* é fazer com que seja possível por parte do computador, a solução de problemas intuitivos ao ser humano, como reconhecimento de fala ou faces. O objetivo é permitir que o computador aprenda através da experiência obtida e entenda conceitos do mundo a partir de uma hierarquia de conceitos, onde cada conceito é definido de acordo com seu relacionamento com outros conceitos mais simples. Através da obtenção de conhecimento por meio da experiência, essa abordagem não necessita da intervenção humana para especificar todo o conhecimento que o computador precisa. Ao se desenhar um gráfico, mostrando como os conceitos são construídos em relação aos outros, obtém-se várias camadas. Daí surge a expressão ***deep*** *learning* (GOODFELLOW et al., 2016).

De acordo com LeCun et al. (2015) as técnicas convencionais de aprendizado de máquina são limitadas em sua habilidade de processar dados na forma bruta. Por conta desse fato, construir um sistema de reconhecimento de padrões ou de aprendizado de máquina requer conhecimento do domínio estudado para se elaborar um extrator de características capaz de transformar os dados da forma bruta para uma representação ou vetor de características que por sua vez possa ser interpretado pelo classificador. Por outro lado, o *deep learning* é um conjunto de métodos que permite ao computador receber dados brutos e descobrir quais representações são necessárias para realizar a detecção ou classificação. Esses métodos se baseiam na aprendizagem de representações com múltiplos níveis de representações. Pequenos módulos são responsáveis por realizar a transformação da representação em um determinado nível (iniciando pelo nível com os dados brutos) para uma representação em um nível superior, um pouco mais abstrato. Dessa maneira, após um número suficiente de transformações, é possível o aprendizado de funções altamente complexas.

As redes baseadas em *deep learning* vem sendo aplicadas com sucesso nos últimos anos para tarefas de classificação (AHMED et al., 2008) e (BENGIO et al., 2007), regressão (SALAKHUTDINOV; HINTON, 2008), redução de dimensionalidade (SALAKHUTDINOV; HINTON, 2007), modelagem de texturas (OSINDERO; HINTON, 2008), recuperação de informação (RANZATO; SZUMMER, 2008) e processamento de linguagem natural (MNIH;

HINTON, 2009).

3.3.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) podem ser consideradas como um tipo de RNA que utiliza uma arquitetura especial, particularmente bem adaptada para o processo de classificação de imagens. Ao se fazer uso dessa arquitetura, é possível treinar redes com muitas camadas, as quais são muito boas para tal tipo de classificação (NIELSEN, 2015).

De acordo com Goodfellow et al. (2016) as CNNs são redes voltadas ao processamento de dados que possuam uma topologia do tipo grade, como dados de imagens, que podem ser considerados como uma grade 2D de pixels. Ainda segundo o autor, a referência do nome *"convolutional neural network"* indica que a rede faz uso de uma operação matemática chamada convolução (*convolution*). Logo, CNNs são redes neurais que fazem uso da convolução para a multiplicação de matrizes em pelo menos uma de suas camadas.

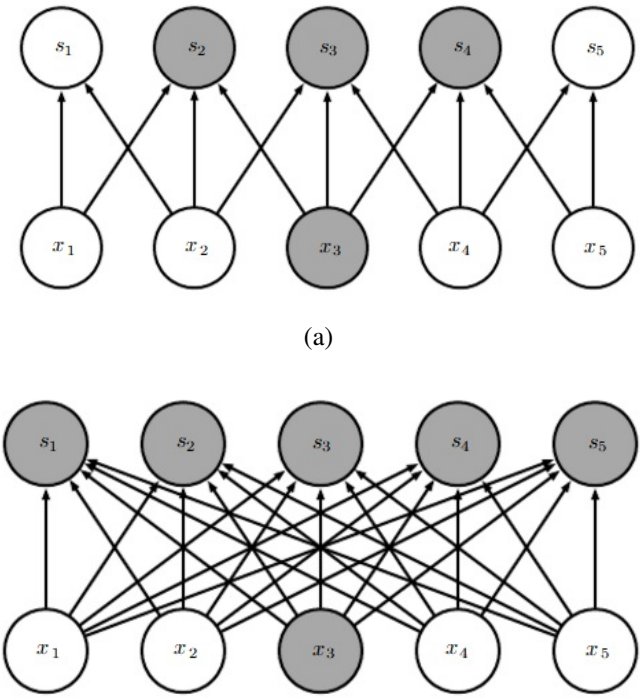
Em sua forma mais generalizada, a convolução é uma operação aplicada a duas funções com o objetivo de se obter uma terceira que retorna um valor estimado. A convolução pode ser denotada pela equação 27, onde $x(t)$ representa uma função de entrada no instante t, enquanto que w representa uma função de média ponderada e a denota uma medida realizada em outro momento diferente de t. Na terminologia das CNNs, o primeiro argumento é nomeado como *entrada* e o segundo como *kernel*, e a saída é geralmente chamada de *mapa de características* (GOODFELLOW et al., 2016).

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a) \tag{27}$$

O processo de convolução alavanca três importantes conceitos que podem auxiliar na melhora de um sistema de aprendizado de máquina: interações esparsas, compartilhamento de parâmetros e representações equivariantes.

- **Interações esparsas:** Nas RNAs tradicionais, as camadas utilizam multiplicação de matrizes por uma segunda matriz de parâmetros com o objetivo de descrever a interação entre cada unidade de entrada com cada unidade de saída. No caso das CNNs, há as interações esparsas, que são alcançadas ao fazer com que o kernel seja menor que a entrada. Ao se processar uma imagem por exemplo, a entrada de dados pode conter milhões de pixels, porém é possível detectar pequenas características como contorno

por meio de kernels que ocupam uma pequena porção desses pixels. Dessa maneira, é necessário armazenar menos parâmetros, o que reduz a quantidade de memória e processamento empregados (GOODFELLOW et al., 2016). A Figura 15(a) ilustra uma interação esparsa onde a convolução é formada por um kernel de tamanho 3, onde apenas 3 saídas *s* são afetadas por *x*. Na Figura 15(b) demonstra o que ocorre quando a conectividade não é esparsa, onde todas as saídas *s* são afetadas por *x*;



(a)

(b)

Figura 15 – Interações esparsas.
Fonte: (GOODFELLOW et al., 2016)

- **Compartilhamento de parâmetros:** A ideia é fazer uso de um mesmo parâmetro para mais de uma função dentro de um modelo. Em uma RNA tradicional, cada elemento da matriz de pesos é utilizado exatamente uma vez ao computar a saída de uma camada, é multiplicado por um elemento da entrada e nunca mais é visitado. Em uma CNN cada membro do kernel é usado em cada posição dos dados de entrada (com algumas exceções com relação aos pixels de fronteira). O compartilhamento de parâmetros utilizado em uma CNN significa que ao invés do sistema aprender um conjunto separado de parâmetros para cada coordenada, ele aprende apenas um único conjunto. Embora isso não afete o tempo da propagação, reduz consideravelmente o armazenamento necessário para o modelo para o número de parâmetros utilizados (GOODFELLOW et al., 2016);
- **Representações equivariantes:** A equivariância é uma propriedade das camadas de uma

CNN que ocorre por consequência do compartilhamento de parâmetros. Uma função ser equivariante, significa que ao se alterar a entrada de dados, a saída deve mudar da mesma maneira. O processo de convolução não é naturalmente equivariante para alguns outros tipos de transformações, como alterações na escala ou rotação de uma imagem, sendo necessários outros mecanismos para tratar essas transformações (GOODFELLOW et al., 2016).

De acordo com Goodfellow et al. (2016) uma camada típica de uma CNN consiste em três estágios. No primeiro, a camada executa diversas convoluções em paralelo para produzir um conjunto de ativações lineares. No segundo estágio, cada ativação é executada através de uma função de ativação não-linear. No terceiro e último estágio, ocorre a utilização de uma função de ***pooling*** para modificar a saída da camada mais distante.

As camadas de *pooling* são utilizadas logo após as camadas de convolução, e tem como finalidade simplificar a informação na saída da camada de convolução. A camada de *pooling* pega cada saída do mapa de características da camada de convolução e prepara um mapa de características mais compacto. Pode-se tomar como exemplo uma situação onde cada neurônio da camada de *pooling* equivale a uma região de tamanho $2x2$ neurônios da camada anterior. Um procedimento comum para a realização do *pooling* é conhecido como max-pooling. A ideia é ter como resultado da saída a ativação máxima de uma região de $2x2$ neurônios. Portanto, considerando um exemplo em que se tenha uma saída com $24x24$ neurônios, após o processo de *pooling* se obtêm $12x12$ neurônios, como ilustrado na Figura 16 (NIELSEN, 2015).

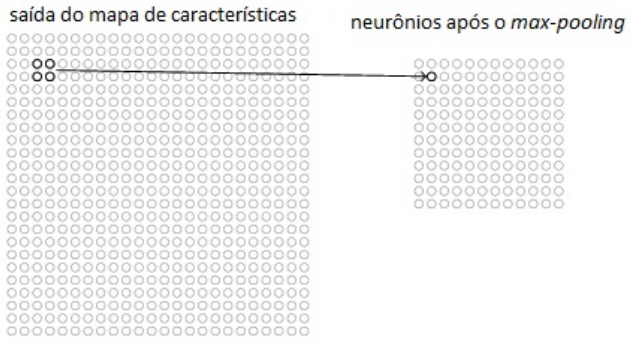


Figura 16 – Processo de pooling.
Fonte: Nielsen (2015)

4 MATERIAIS E MÉTODOS

Nesse capítulo será descrita a metodologia utilizada para o desenvolvimento deste projeto. Serão apresentadas as etapas do projeto e os principais fundamentos e tecnologias a serem empregados.

4.1 MATERIAIS

Nesta seção serão descritos os materiais necessários para a execução do trabalho, tal como o *software* e *hardware* a serem utilizados.

4.1.1 Hardware

- **Computador:** Foram utilizados dois computadores ao longo do desenvolvimento do trabalho: Notebook (Intel Core i5 3230M 2.60GHz, 8GB de memória RAM, GPU AMD Radeon HD8850M 2GB) e um Desktop (Intel Core i7 3770K 3.5GHz, 16GB de memória RAM, GPU NVidia GTX 970 4GB)

4.1.2 Softwares e Linguagens

- **Sistema Operacional:** Responsável pelo gerenciamento dos recursos do sistema e por realizar a interação entre o usuário e o computador. Foi feito uso do Microsoft Windows 8.1[1];

- **Linguagem de programação: Python 3.5[2]:** é uma linguagem interpretada orientada a objetos multi-plataforma, podendo ser executada em diversos sistemas operacionais. Essa linguagem foi escolhida para ser utilizada em alguns processos das etapas de processamento de imagens e principalmente para a aplicação dos métodos de classificação com *deep learning* devido ao fato de ser amplamente utilizada para esse fim, por conta de algumas bibliotecas que auxiliam no processo;

- **Ambiente de Desenvolvimento: Anaconda[3]:** trata-se de uma distribuição para desenvolvimento de aplicações na linguagem Python, voltado principalmente para a área de *data science*, tendo em vista que possui diversas bibliotecas voltadas a este fim, além de contar com um gerenciador de ambientes virtuais;

- **TensorFlow[4]:** é uma biblioteca em código aberto utilizada para computação de cálculos em alta performance. Possui uma arquitetura que permite a implantação de aplicações para diversas plataformas, desde desktops e clusters até dispositivos móveis;

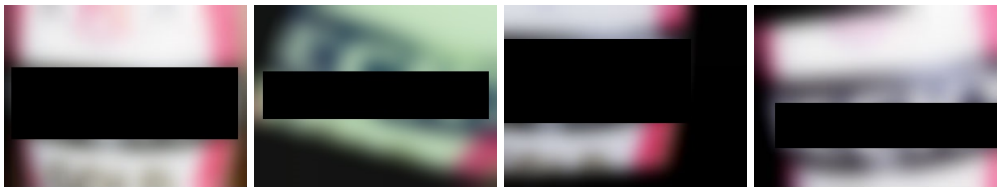
- **Keras[5]:** é uma API para *deep learning* escrita em Python e que é capaz de rodar em conjunto com o TensorFlow ou outras bibliotecas voltadas ao mesmo propósito. Permite a construção de redes neurais de maneira rápida e descomplicada, o que permite realizar a prototipação de um trabalho em pouco tempo.

[1] https://www.microsoft.com/pt-br/windows
[2] https://www.python.org/
[3] https://www.anaconda.com
[4] https://www.tensorflow.org
[5] https://www.keras.io

4.1.3 Base de Imagens

Para a realização dos testes ao longo do desenvolvimento do trabalho foi feito uso de uma base de logotipos já existente, conhecida como FlickrLogos-32[6] Romberg et al. (2011), a qual conta com imagens contendo logotipos de 32 diferentes marcas, para o trabalho foram utilizadas imagens de 10 dessas marcas. A qualidade das imagens ou as situações em que podem ser apresentadas podem variar, visando assim verificar a capacidade de detecção e classificação em situações onde a imagem não segue um padrão específico.

Como a quantidade de imagens por marca contidas na base FlickrLogos-32 não é o suficiente para se alcançar o mínimo de imagens planejadas, foi realizado um processo de *Data Augmentation*, que consiste em gerar novas imagens baseadas nas imagens já existentes na base. O processo foi realizado por meio de uma biblioteca escrita em Python, chamada *imgaug*[7].



(a) (b) (c) (d)

Figura 17 – (a) Imagem original, (b), (c) e (d) Novas imagens geradas.
Fonte: Autoria própria

É possível verificar na Figura 17 alguns exemplos de imagens geradas a partir de outra. O *imgaug* permite a definição de diversos parâmetros para restringir o que e em qual proporção pode ser alterado nas imagens resultantes, tais como área de corte, ângulo, coloração, dentre outros.

Cada uma das 10 classes contava com 70 imagens, sendo que foram geradas novas 64 imagens para cada uma, totalizando 4550 imagens para cada classe, sendo assim 45500 ao total.

[6] http://www.multimedia-computing.de/flickrlogos/
[7] https://github.com/aleju/imgaug

4.2 MÉTODO

O trabalho se divide em duas etapas, sendo a primeira a realização do treinamento de uma CNN para classificação de rótulos, e em seguida, utilizando a rede previamente treinada, elaborar um sistema CBIR fazendo uso das características extraídas pela própria rede durante o seu treinamento.

4.2.1 Treinamento da CNN

Foram testadas algumas estruturas de rede para o problema proposto, tais como Alexnet, VGG e LeNet, e foi optado por utilizar um modelo utilizado previamente na classificação da base de imagens CIFAR-10[8], principalmente por sua simplicidade, que torna a alteração da estrutura mais fácil de ser realizada. O modelo é composto por quatro camadas convolucionais e duas camadas totalmente conectadas. Cada *kernel* segue um padrão *3x3*, e cada camada convolucional possui apenas 32 ou 64 filtros. Na base de imagens CIFAR-10, onde há uma generalização maior, tendo em vista que contém classes muito distintas umas das outras, como carros e animais, o modelo foi capaz de alcançar resultados na casa dos 95%. A estrutura do modelo é representada na Figura 18. Além das camadas convolucionais já citadas anteriormente, é possível verificar a presença de camadas de normalização e ativação.

Como comentado na Seção 4.1 a base de imagens para o trabalho conta com 10 classes, cada uma contendo 4550 imagens. Para o treinamento da rede é necessário fazer uma divisão das imagens, uma parte para treino e uma segunda parte que é utilizada pela rede para realizar a validação provisória do modelo enquanto a rede é treinada. Para tal foi definida uma proporção 80/20, o que resultou em 3640 imagens para treino e 910 imagens para o processo de validação. Como resultado se tem duas pastas (treino e validação), sendo que dentro de cada uma dessas, as imagens são separadas de acordo com a classe correspondente, em pastas nomeadas *C0, C1, C2,..., C9*. Essa divisão é necessária por parte do Keras, para que se possa carregar as imagens na memória em forma de um vetor de dados, que posteriormente serve para alimentar a camada de entrada da CNN. No momento antes das imagens serem repassadas para a rede, as mesmas são redimensionadas para um tamanho de 64x64 pixels. Embora haja uma pequena

[8]https://www.cs.toronto.edu/ kriz/cifar.html

Figura 18 – Representação da estrutura da CNN utilizada.

Fonte: Autoria própria.

perda de qualidade e de informação visual ao realizar esse redimensionamento, esse processo torna o treinamento da rede menos custoso, o que reduz de maneira considerável o tempo de treinamento e os recursos de hardware necessários.

Após a finalização das etapas que envolvem o treinamento da rede, o modelo é salvo, tornando assim possível a utilização do mesmo para a classificação de novas imagens até então desconhecidas da rede. O modelo salvo é também utilizado para a segunda parte do trabalho, o sistema CBIR.

4.2.2 CBIR utilizando as características extraídas pela CNN

Como foi visto na Seção 2, um sistema de recuperação de imagem baseado em conteúdo é composto por algumas etapas, sendo uma das mais importantes a extração de características das imagens. Em sistemas desse tipo, a escolha de bons descritores para o problema proposto é um ponto crucial para determinar a eficiência do mesmo(KUMAR et al., 2012). Contudo, na grande maioria das vezes, os descritores são escolhidos por meio de testes, analisando a eficácia de cada um ou de uma combinação dos mesmos. Considerando a atual capacidade das CNNs de realizar a classificação de imagens com eficiência beirando os 99% em muitos casos, surge a possibilidade de se fazer uso das características obtidas pela mesma para o uso em outras aplicações, evitando assim o processo de seleção de descritores que sejam eficientes para um caso em específico.

Embora esteja sendo avaliado para um tarefa de classificação, é importante ressaltar que a possibilidade de se extrair características de um modelo CNN torna possível a utilização das mesmas em outros sistemas, assim como o CBIR, que pode ser utilizado para fins onde haja a necessidade de se realizar uma busca onde não há dados rotulados ou mesmo para encontrar semelhanças visuais dentro de uma base de imagens.

Fazendo uso do modelo treinado previamente, foi criado um novo modelo, retirando as últimas camadas responsáveis pela parte de classificação do modelo. Dessa maneira, se tem em mãos uma rede que recebe imagens, e ao invés de realizar a classificação, retorna um vetor contendo informações referentes às características dessas imagens.

A Figura 19 exemplifica o que foi feito na estrutura da rede, as quatro últimas camadas, responsáveis por realizar a classificação foram removidas. Dessa maneira é possível recuperar a saída da camada *Dense*, que é equivalente a uma camada totalmente conectada. Para cada imagem que a rede recebe, essa camada gera um vetor de saída contendo 512 posições, sendo essas informações referentes à características da imagem que foram extraídas pela rede ao longo do treinamento.

A base de imagens utilizada no treinamento da rede é carregada e armazenada em um vetor, a fim de ser utilizada posteriormente na recuperação de imagens. É então realizada uma predição com a rede fazendo uso dessas imagens, porém como a estrutura foi alterada, ao invés de se obter uma das 10 possíveis classes, se obtém um uma matriz no formato 3640x512, referente ao número de imagens e as características extraídas de cada uma.

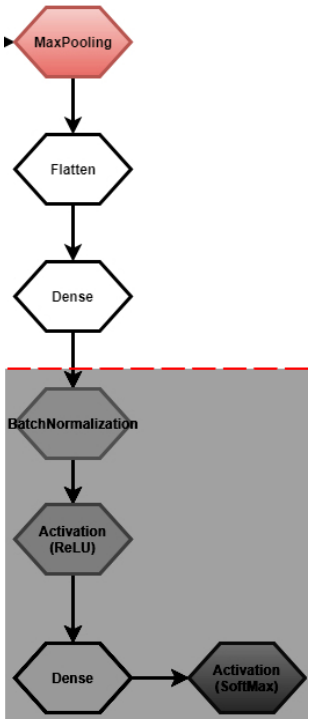


Figura 19 – Representação das camadas retiradas da CNN.

Fonte: Autoria própria.

4.2.2.1 Principal Component Analysis

Após se obter a saída da rede se tem em mãos uma saída de 512 valores para cada imagem da base, porém, é interessante realizar uma análise com o objetivo de verificar se há a necessidade dessa quantidade de informação. Quanto maior a quantidade de descritores, maior a dimensionalidade do problema, e consequentemente o tempo e o custo computacional aumentam de maneira exponencial.

O PCA (*Principal Component Analysis*) é um método de transformação linear ortogonal que transforma os dados em um novo sistema de coordenadas, de maneira que a maior projeção dos dados chegue à primeira coordenada (nomeada primeiro componente principal), a segunda maior variação na segunda coordenada e assim sucessivamente (NETO, 2013). Quando utilizado com a finalidade de se reduzir a dimensionalidade, o processo envolve zerar um ou mais dos menores componentes principais encontrados por meio da análise, resultando assim em uma projeção dos dados com uma dimensão menor do que a original, porém mantendo a relevância dos dados.

A Figura 20 ilustra o quanto a redução pode afetar as informações contidas em uma

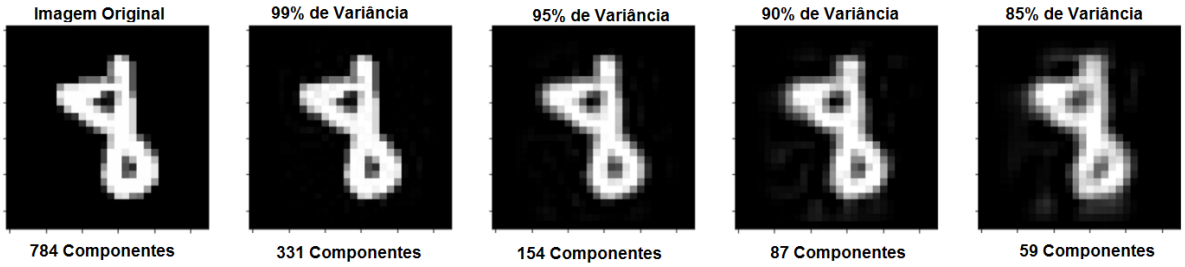


Figura 20 – Exemplo de uma imagem com diferentes quantidades de informação retida.
Fonte: (GALARNYK, 2017)

imagem. Pode-se notar que ao utilizar 85% da variância, a imagem, visualmente perde um pouco da informação, porém continua sendo possível identificar a mesma, por outro lado o número de componentes é apenas 7,5% com relação ao original (784 componentes na imagem original e 59 na reduzida).

Para o trabalho foi realizada uma análise considerando 85% da variância. Na Figura 21 é possível notar o comportamento da curva, que ao chegar próximo da casa dos 90%, sofre alterações muito pequenas mesmo com um grande aumento na quantidade de componentes utilizados. Dessa maneira, ao invés dos 512 valores iniciais, foram utilizados 21 valores retornados pelo PCA.

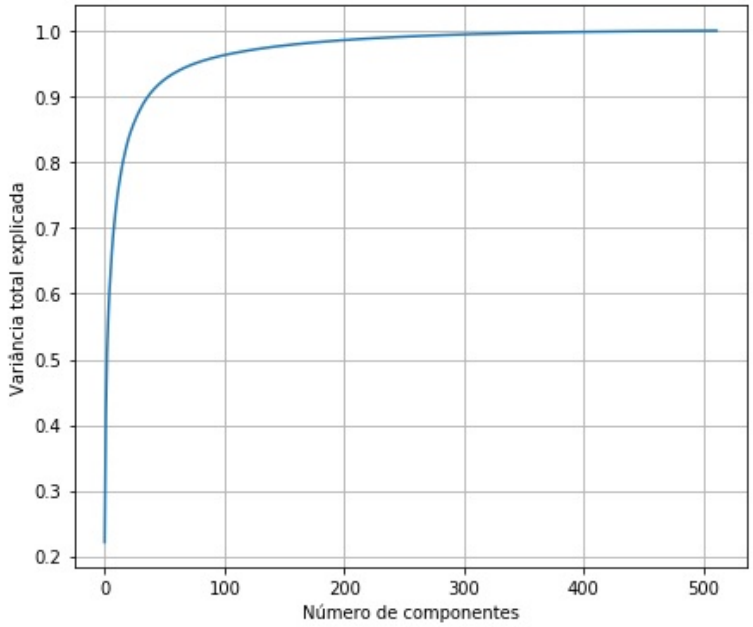


Figura 21 – Gráfico da relação entre o número de componentes e a variância.
Fonte: Autoria própria.

4.2.2.2 Treinamento com kNN

Como o objetivo do sistema CBIR é retornar as imagens semelhantes, foi optado por se fazer uso de um algoritmo kNN (*k-Nearest Neighbors*), o qual tem como função retornar os vizinhos mais próximos dado um determinado ponto. Previamente foram feitos testes sem a utilização do kNN, fazendo uso apenas de uma função que realizava o calculo da distância entre o vetor da imagem fornecida com os vetores das imagens de toda a base, e então organizava um ranking com os mais próximos, contudo o processo é extremamente lento, o que o torna inviável para muitas situações.

O kNN é alimentado com os dados resultantes do PCA, e foi definido a busca pelos 4 vizinhos mais próximos para cada imagem fornecida para teste.

Todos os passos descritos desde a extração de características até o kNN são utilizados tanto para a base de treinamento quanto para a imagem que for fornecida ao sistema pelo usuário. Por fim, para avaliar a eficácia tanto do modelo CNN, como do sistema CBIR, foram adquiridas novas 20 imagens, sendo duas de cada classe, para a realização dos testes. As imagens podem ser vistas na Figura 22, e passaram também pela etapa de redimensionamento para o tamanho de 64x64 para estar de acordo com a entrada da rede. Todas as imagens apresentadas foram desfocadas para apresentação no trabalho, com a intenção de não infringir direitos das marcas.

A fim de se fazer uma comparação, foi feita uma versão do CBIR fazendo uso de algumas das características comentadas na sessão 2.2, no caso, matriz de co-ocorrência e histograma de cores, essas extraídas manualmente, apenas com a intenção de demonstrar o quão complexo pode ser essa tarefa quando realizada de maneira habitual.

Figura 22 – Imagens utilizadas para teste do modelo e do sistema CBIR.

Fonte: Autoria própria.

5 RESULTADOS

Nesse capítulo serão exibidos os resultados dos experimentos realizados, tal como a discussão sobre os mesmos.

5.1 MODELO CNN

Enquanto realiza-se o treinamento da rede, pode-se verificar uma estimativa da acurácia da mesma, que é calculada baseando-se na parte da base que foi separada para validação. Após 20 épocas, a rede apresentou uma taxa de acerto de 98,7%. A duração do treino foi de apenas 17 minutos fazendo uso de uma GPU com 1024 núcleos.

Foi analisada a eficácia da rede com as 20 imagens selecionadas para a etapa de teste, onde foi atingido a marca de 100% na taxa de acerto. A Figura 23 mostra a matriz de confusão dos resultados obtidos durante o teste. No eixo vertical estão as classes corretas, enquanto que no eixo horizontal encontram-se as predições realizadas pela rede.

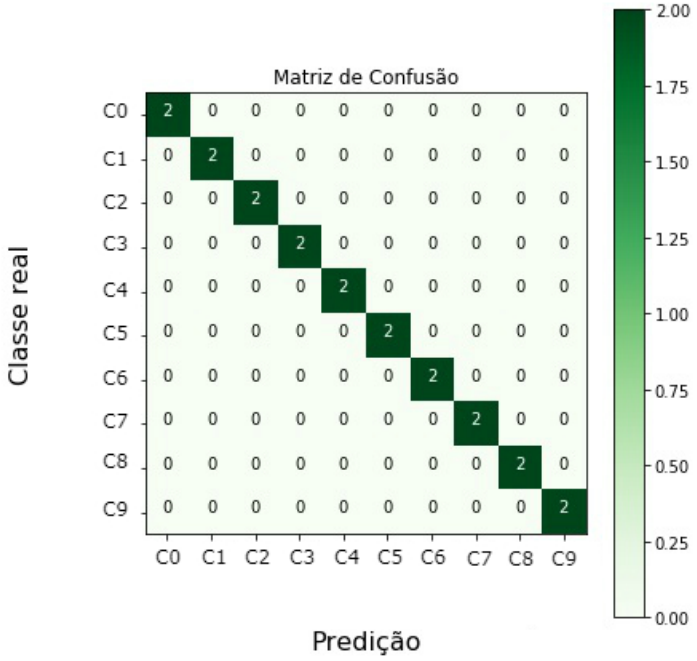


Figura 23 – Matriz de confusão com os resultados dos testes.
Fonte: Autoria própria.

O modelo demonstra ser muito eficiente para o problema proposto, mesmo possuindo uma estrutura consideravelmente simples e que pode ser treinada em poucos minutos. É importante ressaltar que o modelo é eficiente considerando as imagens que possuem apenas a área do rótulo, tendo em vista que a rede foi treinada dessa maneira.

5.2 SISTEMA CBIR

Para a análise do CBIR, é necessário testar as imagens individualmente e verificar se os resultados retornados são compatíveis com os esperados. Como foi comentado na Sessão 4.2, foi optado por se retornar os 4 vizinhos mais próximos por meio do kNN, dessa forma as 4 imagens que o algoritmo julgar mais semelhantes devem ser retornadas. A sequência de figuras a seguir mostra os resultados obtidos para cada uma das 20 imagens utilizadas para teste. A primeira imagem de cada linha representa a imagem de teste, e as 4 imagens seguintes os resultados trazidos da busca realizada pelo sistema.

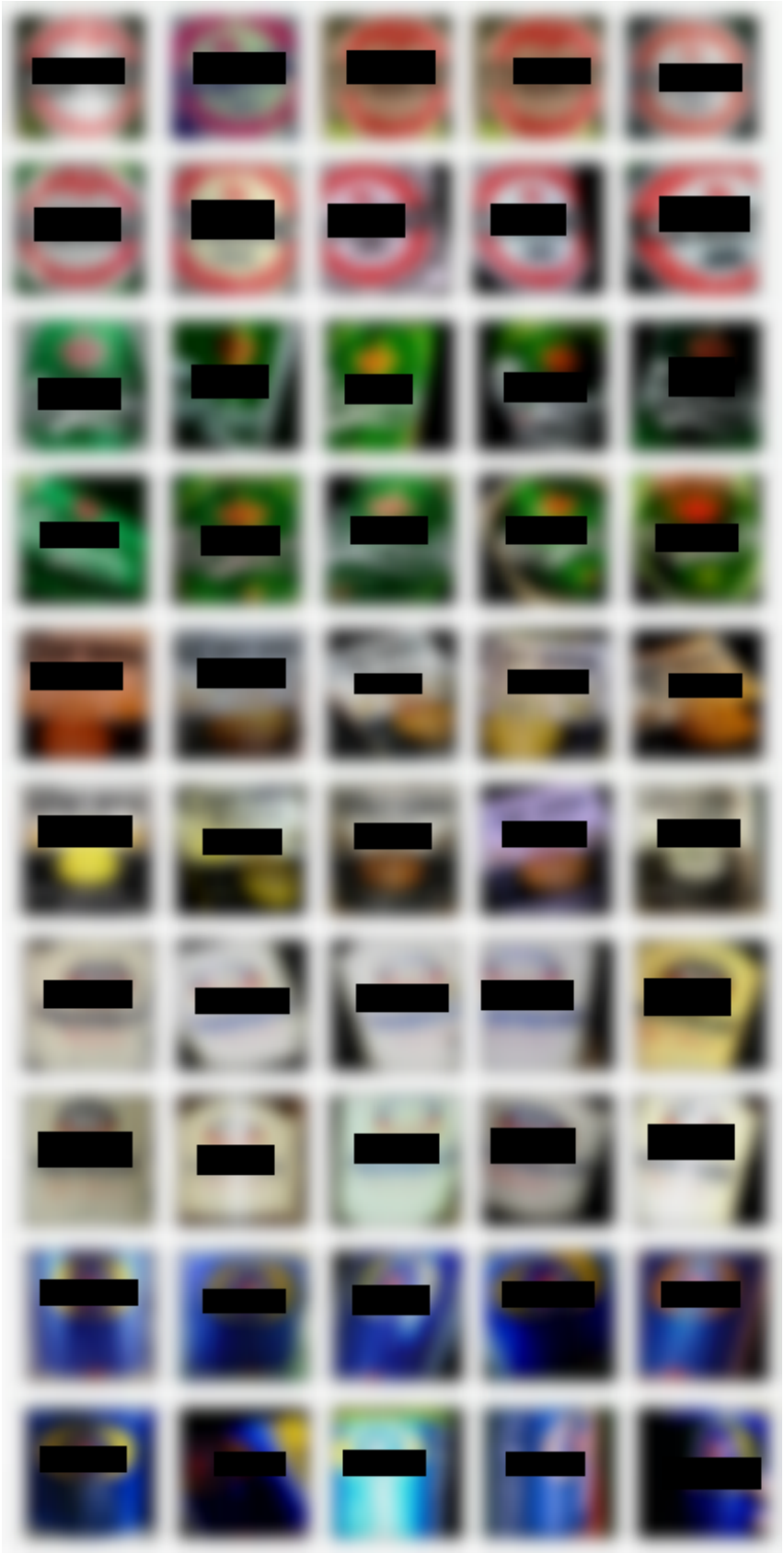


Figura 24 – Resultados do CBIR para as classes C0, C1, C2, C3 e C4.

Fonte: Autoria própria.

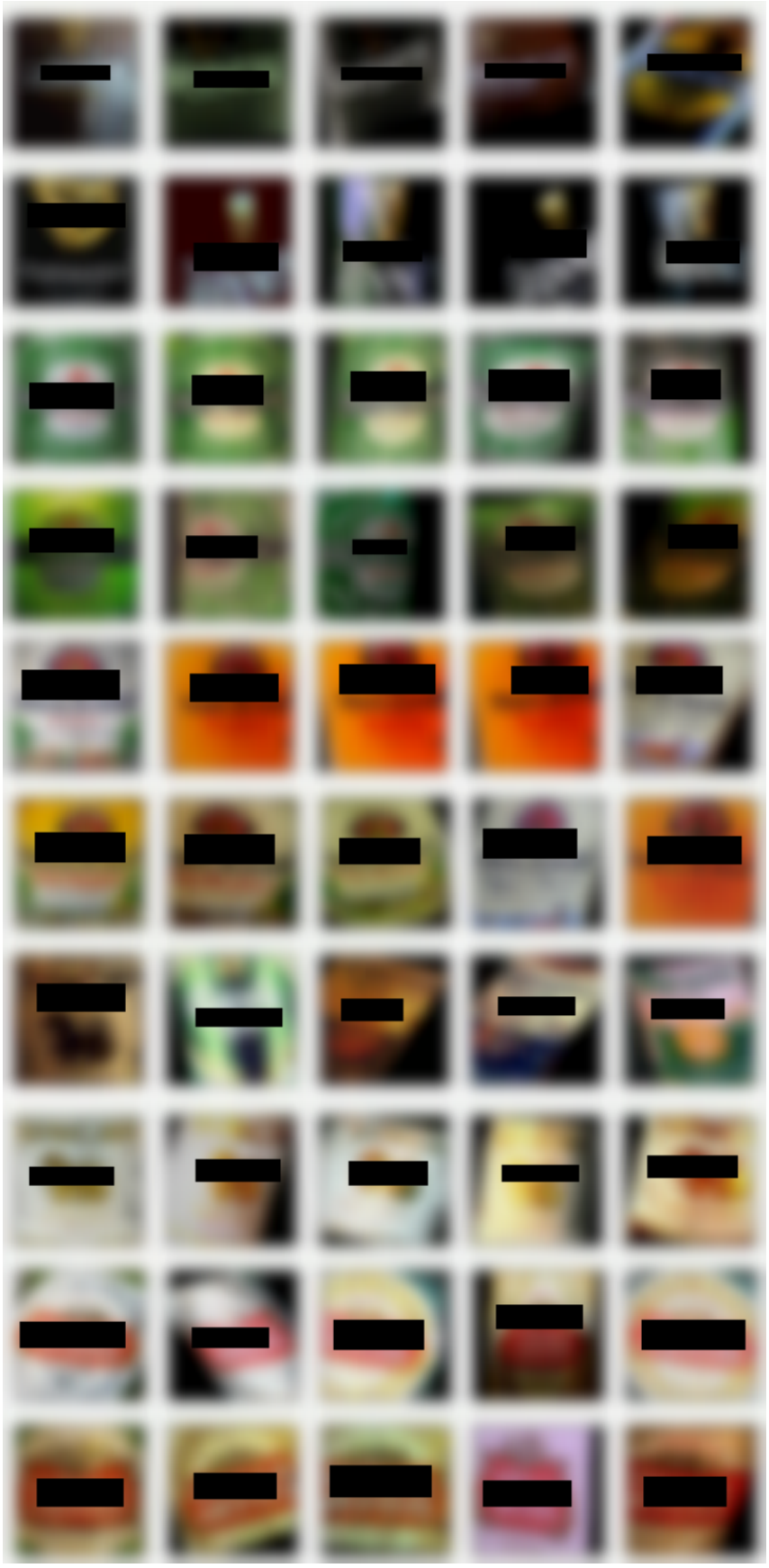


Figura 25 – Resultados do CBIR para as classes C5, C6, C7, C8 e C9.

Fonte: Autoria própria.

Analisando os resultados obtidos pelo CBIR nas Figuras 24 e 25, em apenas um dos casos as imagens retornadas não correspondem à marca da imagem fornecida. Para as demais imagens, não só a primeira imagem retornada é correta, mas sim todas as quatro. Se fosse analisar o sistema para uma tarefa de classificação, baseado no teste realizado com as 20 imagens, significaria uma taxa de acerto de 95%.

Foram realizados alguns testes fazendo uso de características extraídas por conta própria, sem uso da CNN. O método empregado foi semelhante, utilizando a mesma estrutura do CBIR e o kNN para retorno dos resultados, além disso foi feito uso dos 6 descritores da Matriz de Co-ocorrência propostos por Baraldi e Parmiggiani (1995), que são energia, entropia, contraste, variância, correlação e homogeneidade. Também foi realizado um teste fazendo uso de um descritor baseado em cor, no caso um histograma.

Figura 26 – Exemplo de resultados utilizando Histograma de Cor como descritor.
Fonte: Autoria própria.

É possível observar na Figura 26 que ao se fazer uso do histograma de cor como descritor, pode se obter resultados positivos em alguns casos, porém é pouco robusto pelo fato de se basear apenas na coloração das imagens. Já os resultados da Figura 27 demonstram que, para o problema proposto, a análise por meio da matriz de co-ocorrência por si só é pouco eficiente.

Figura 27 – Exemplo de resultados utilizando matriz de co-ocorrência.
Fonte: Autoria própria.

Existe a possibilidade de se utilizar diversos outros descritores e até mesmo realizar uma combinação dos mesmos na tentativa de se obter bons resultados para o problema em questão. O ponto principal desses testes complementares é deixar claro que para problemas específicos há a necessidade de se fazer uso de descritores específicos, o que torna a tarefa de seleção das características a serem extraídas altamente complexa.

6 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Neste trabalho foi apresentado um método eficiente para se fazer uso das características extraídas por meio de uma rede CNN modificada em um sistema de recuperação de imagens baseado em conteúdo. O método utilizado, além de apresentar bons resultados, é relativamente mais prático em relação ao processo de extração de características usual, uma vez que não há a necessidade de se analisar quais descritores serão utilizados, tendo em vista que a própria CNN recupera as informações relevantes das imagens para o processo de classificação. Outro ponto importante é que mesmo tratando-se de *deep learning*, o treinamento da rede com o modelo utilizado é consideravelmente rápido se comparado com o processo de extração de características realizado para os testes adicionais apresentados.

Embora haja a necessidade de se considerar que para casos específicos, um sistema CBIR possa apresentar melhores resultados quando feito com descritores especialmente selecionados, o método apresentado nesse trabalho se demonstrou muito eficiente e promissor para tarefas desse segmento.

Como sugestão para trabalhos futuros fica a utilização do método proposto para outros segmentos como por exemplo a área médica, realizando o treinamento de uma rede voltada a esse propósito e assim aplicar os princípios discutidos ao longo do trabalho a fim de verificar a eficácia do mesmo. Outro ponto interessante a se pensar, é em uma implementação voltada a dispositivos móveis, visto que atualmente os *frameworks* voltados para trabalho com *deep learning* possuem versões otimizadas para esse segmento, o que possibilita a implementação de um sistema utilizando a mesma metodologia aplicada no trabalho.

REFERÊNCIAS

AHMED, A.; YU, K.; XU, W.; GONG, Y.; XING, E. Training hierarchical feed-forward visual recognition models using transfer learning from pseudo-tasks. **Proceedings of the 10th European Conference on Computer Vision (ECCV'08)**, p. 69–82, 2008.

BARALDI, A.; PARMIGGIANI, F. An Investigation of the Textural Characteristics Associated with Gray Level Cooccurrence Matrix Statistical Parameters. **IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING**, p. 293–304, 1995.

BENGIO, Y.; LAMBLIN, P.; POPOVICI, D.; LAROCHELLE, H. Greedy layer-wise training of deep networks. **Advances in Neural Information Processing Systems 19 (NIPS'06)**, p. 153–160, 2007.

BISHOP, C. M. **Pattern Recognition and Machine Learning**. 1st. ed. New York: Springer, 2006.

BRADSKI, G.; KAEHLER, A. **Learning OpenCV**. 1st. ed. Sebastopol: O'Reilly, 2008.

BRAGA, A. de P.; CARVALHO, A. P. de Leon F. de; LUDERMIR, T. B. **Redes Neurais Artificiais: Teoria e Aplicações**. 1st. ed. Rio de Janeiro: LTC Editora S.A., 2000.

BROWN, L. **Deep Learning For Image Classification**. 2015. Disponível em: <http://www.nvidia.com/content/events/geoInt2015/LBrown_DL_Image_Classific ationGEOINT.pdf>. Acesso em: 29 de outubro de 2016.

BROWNLEE, J. **Basic Concepts in Machine Learning**. 2015. Disponível em: <http://machinelearningmastery.com/basic-concepts-in-machine-learning/>. Acesso em: 30 de setembro de 2016.

BUGATTI, P. H.; KASTER, D. S.; PONCIANO-SILVA, M.; TRAINA, C.; AZEVEDO-MARQUES, P. M.; TRAINA, A. J. M. PRoSPer: Perceptual similarity queries in medical CBIR systems through user profiles. **Computers in Biology and Medicine**, v. 45, n. 1, p. 8–19, 2014. ISSN 00104825.

CONCI, A.; AZEVEDO, E.; LETA, F. R. **Computação Gráfica**. 1st. ed. São Paulo: Campus/Elsevier, 2008.

Digital For All Now. **MACHINE LEARNING: FROM READING A HANDWRITTEN ADDRESS TO PREDICTING THE ITEMS IN A SHOPPING BASKET**. 2015. Disponível em: <https://www.digitalforallnow.com/en/machine-learning-predictive-model-business-intelligence/>. Acesso em: 24 de setembro de 2016.

DOMINGOS, P. A few useful things to know about machine learning. **Commun. ACM**, ACM, New York, NY, USA, v. 55, n. 10, p. 78–87, out. 2012. ISSN 0001-0782. Disponível em: <http://doi.acm.org/10.1145/2347736.2347755>.

FACELI, K.; LORENA, A. C.; GAMA, J.; CARVALHO, A. C. P. de Leon Ferreira de. **Inteligência Artificial: Uma abordagem de Aprendizado de Máquina**. 1st. ed. Rio de Janeiro: LTC Editora Ltda., 2011.

GALARNYK, M. **PCA using Python**. 2017. Disponível em: <https://towardsdatascience.com/pca-using-python-scikit-learn-e653f8989e60>. Acesso em: 13 de maio de 2018.

GONZALEZ, R. C.; WOODS, R. E. **Processamento de Imagens Digitais**. 1st. ed. São Paulo: Edgard Blücher Ltda, 2000.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. Deep learning. Book in preparation for MIT Press. 2016. Disponível em: <http://www.deeplearningbook.org>.

GUAN, L.; HE, Y.; KUNG, S.-Y. **Multimedia Image and Video Processing**. 2nd. ed. New York: CRC Press - Taylor e Francis Group, 2012.

HARALICK, R. M.; SHANMUGA, K.; DINSTEIN., I. Textural features for image classification. p. 610–621, 1973.

HAYKIN, S. **Redes Neurais: Princípios e Prática**. 2nd. ed. Porto Alegre: Bookman, 2001.

HELFER, G. A.; FERRãO, M. F.; FERREIRA, C. de V.; HERMES, N. Pulicação de métodos de análise multivariada no controle qualitativo de essências alimentícias empregando espectroscopia no infravermelho médio. **Food Science and Technology (Campinas)**, scielo, v. 26, p. 779 – 786, 12 2006. ISSN 0101-2061. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0101-20612006000400011&nrm=iso>.

HERNANDEZ, E. **Is it possible to create a computer that mimics human intelligence by replicating the way the human brain processes information?** 2011. Disponível em: <https://www.nshss.org/media/1533/hernandez.pdf>. Acesso em: 30 de outubro de 2016.

KHOKHER, A.; GOBINDGARH, M. Content-based Image Retrieval : Feature Extraction Techniques and Applications. n. March, p. 9–14, 2008.

KNORR, E. **How PayPal beats the bad guys with machine learning**. 2015. Disponível em: <http://www.infoworld.com/article/2907877/machine-learning/how-paypal-reduces-fraud-with-machine-learning.html>. Acesso em: 24 de setembro de 2016.

KUMAR, E. S.; SUMATHI, A.; LATHA, K. Featuer selection and extraction for content-based image retrieval. **International Journal of Mathematics Trends and Technology**, p. 70–73, 2012.

KUNZMAN, D. **Understanding What a Histogram NOT is Telling You**. 2016. Disponível em: <http://davidkunzman.net/photographyArticles/histograms/part2.php>. Acesso em: 21 de setembro de 2016.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. n. 521, p. 436–444, 2015.

LUX, M. Content Based Image Retrieval. **Main**, v. 28, p. 735–738, 2011. Disponível em: <http://www.jisc.ac.uk/uploaded_documents/jtap-039.>

MITCHELL, T. M. The Discipline of Machine Learning. 2006.

MNIH, A.; HINTON, G. E. A scalable hierarchical distributed language model. **Advances in Neural Information Processing Systems 21 (NIPS'08)**, p. 1081–1088, 2009.

MURKANE, K. **Thirteen Companies That Use Deep Learning To Produce Actionable Results.** 2016. Disponível em: <http://www.forbes.com/sites/kevinmurnane/2016/04/01/thirteen-companies-that-use-deep-learning-to-produce-actionable-results/>. Acesso em: 29 de outubro de 2016.

NETO, J. **Principal Component Analysis**. 2013. Disponível em: <http://www.di.fc.ul.pt/~jpn/r/pca/pca.html>. Acesso em: 25 de junho de 2018.

NIELSEN, M. A. Neural networks and deep learning. 2015. Disponível em: <http://neuralnetworksanddeeplearning.com>.

OSINDERO, S.; HINTON, G. E. Modeling image patches with a directed hierarchy of markov random field. **Advances in Neural Information Processing Systems 20 (NIPS'07)**, p. 1121–1128, 2008.

PEDRINI, H.; SCHWARTZ, W. R. **Análise de Imagens Digitais**. st. São Paulo: Thomsom Learning, 2008.

Ponti Jr., M. **Image descriptors: color**. 2011. Disponível em: <http://wiki.icmc.usp.br/images/5/5d/Dip09_imagedescription-color.pdf>. Acesso em: 21 de setembro de 2016.

RANZATO, M.; SZUMMER, M. Semi-supervised learning of compact document representations with deep networks. **Proceedings of the Twenty-fifth International Conference on Machine Learning (ICML'08)**, p. 792–799, 2008.

ROMBERG, S.; PUEYO, L. G.; LIENHART, R.; ZWOL, R. van. Scalable logo recognition in real-world images. In: **Proceedings of the 1st ACM International Conference on Multimedia Retrieval**. New York, NY, USA: ACM, 2011. (ICMR '11), p. 25:1–25:8. ISBN 978-1-4503-0336-1. Disponível em: <http://www.multimedia-computing.de/flickrlogos/>.

SAHBI, H.; BALLAN, L.; SERRA, G.; Del Bimbo, A. Context-dependent logo matching and recognition. **IEEE Transactions on Image Processing**, v. 22, n. 3, p. 1018–1031, 2013. ISSN 10577149.

SALAKHUTDINOV, R.; HINTON, G. E. Learning a nonlinear embedding by preserving class neighbourhood structure. **Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS'07)**, 2007.

SALAKHUTDINOV, R.; HINTON, G. E. Using deep belief nets to learn covariance kernels for gaussian processes. **Advances in Neural Information Processing Systems 20 (NIPS'07)**, p. 1249–1256, 2008.

SCHILS, P. **Color Space Period**. 2010. Disponível em: <http://www.color-theory-phenomena.nl/08.02.html>. Acesso em: 30 de setembro de 2016.

SCURI, A. E. **Fundamentos da Imagem Digital.** 2002. Disponível em: <https://webserver2.tecgraf.puc-rio.br/~scuri/download/fid.pdf>. Acesso em: 13 de setembro de 2016.

SHENGJIU, W. Technical Report, **A Robust CBIR Approach Using Local Color Histograms**. 2001. 7 p.

SHIVHARE, S. Content Based Image Retrieval by Using Interactive Relevance Feedback Technique – A Survey. p. 4641–4645, 2015.

SIMON, P. **Too Big to Ignore: The Business Case for Big Data.** 1st. ed. Hoboken: Wiley, 2013.

SMOLA, A.; VISHWANATHAN, S. **Introduction to Machine Learning**. 1st. ed. Cambridge: Press Syndicate of the University of Cambridge, 2008.

STRICKER, M.; ORENGO, M. Similarity of color images. In: . [S.l.: s.n.], 1995. p. 381–392.

SUGAMYA, K.; PABBOJU, S.; BABU, A. V. A CBIR classification using support vector machines. **2016 International Conference on Advances in Human Machine Interaction (HMI)**, p. 1–6, 2016. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7449193>.

SWAIN, M. J.; BALLARD, D. H. Color indexing. **International Journal of Computer Vision**, v. 7, n. 1, p. 11–32, 1991. ISSN 1573-1405. Disponível em: <http://dx.doi.org/10.1007/BF00130487>.

THEODORIDIS, S.; KOUTROUMBAS, K. **Pattern Recognition**. 4th. ed. San Diego: Academic Press, 2009.

TORRES, R. S.; FALCãO, A. X. Recuperação de Imagens Baseada em Conteúdo Abstract :. **Técnicas e Ferramentas de Processamento de Imagens Digitais e Aplicações em Realidade Virtual e Misturada**, p. 111–132, 2008.

WANG, L.; HE, D.-C. Texture Unit , Texture Spectrum , and Texture Analysis. v. 28, n. 4, p. 509–512, 1990.

WANG, S.; CHAOVALITWONGSE, W.; BABUSKA, R. Machine learning algorithms in bipedal robot control. **IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)**, v. 42, n. 5, p. 728–743, Sept 2012. ISSN 1094-6977.

YU, H.; CAO, J.; LIU, Y.; LUO, W. Non-equal spacing division of hsv components for wood image retrieval. In: **Image and Signal Processing, 2009. CISP '09. 2nd International Congress on**. [S.l.: s.n.], 2009. p. 1–3.

ZHANG, Q.; GOLDMAN, S. A.; YU, W. Content-Based Image Retrieval Using Multiple-Instance Learning. **Proceedings of 19th International Conference on Machine Learning (ICML-2002)**, p. 682–689, 2002.