Implementação de um Sistema de Reconhecimento de Objetos em Imagens

Oeslle A. S. Lucena†, Luciana R. Veloso‡ e Waslon T. A. Lopes§
†Universidade Estadual de Campinas (Unicamp), Campinas, SP
‡Universidade Federal de Campina Grande (UFCG), Campina Grande, PB
§Universidade Federal da Paraíba (UFPB), João Pessoa, PB
e-mails: o190715@unicamp.br, luciana.veloso@dee.ufcg.edu.br e waslon@cear.ufpb.br.

***Resumo*—Um dos passos mais importantes no processamento de dados em sistemas inteligentes é o reconhecimento de objetos. Assim, o estudo de técnicas para reconhecimento de objetos é extremamente importante, desde que é necessário desenvolver sistemas efetivos com altas taxas de reconhecimento que sejam invariantes a rotação, ruído, iluminação, etc. Esta tarefa pode ser definida como a identificação de um objeto individual como membro de uma classe particular que contém objetos com características similares. Para implementar esse tipo de algoritmo são utilizados classificadores e descritores. Neste contexto, este trabalho apresenta um estudo inicial de técnicas de reconhecimento de objetos para validar um sistema de reconhecimento. O algoritmo implementado usa o descritor o histograma de gradiente orientado (HOG – *Histogram of Oriented Gradient*) e uma máquina de vetores de suporte (SVM – *Support Vector Machine*) como classificador e foi avaliado em termos da taxa de reconhecimento considerado três bases de dados: *Caltech-101*, *MSRC v1* e *Stanford Cars Dataset*.**

***Index Terms*—Reconhecimento de objetos, descritores, classificadores, HOG, SVM, taxa de reconhecimento.**

I. Introdução

O uso de imagens digitais vem se expandindo bastante no dia a dia das pessoas. As áreas de biometria, vigilância residencial e comercial, separação de frutas e interação homem-máquina são alguns dos exemplos de atividades que requerem processamento digital de imagens. Para atender essa crescente demanda, cada vez mais, novas tecnologias de visão computacional vem sendo investigadas e desenvolvidas.

Após a aquisição dos dados, um dos mais desafiadores passos para manipulação e processamento dos dados de sistemas inteligentes em visão compútacional é reconhecer objetos [1], [2]. Desse modo, o estudo de técnicas para reconhecer objetos é de extrema importância, uma vez que há a necessidade de desenvolver sistemas eficazes com altas taxas de reconhecimento, que sejam invariantes a rotação, ruído, luminosidade, etc [3]–[5].

Para implementar um algoritmo de reconhecimento de objetos descritores e classificadores são utilizados. Sendo o primeiro responsável por discriminar, de forma compacta, a informação do objeto de interesse, e o segundo, responsável por identificar a qual classe o objeto pertence na imagem, com base nas informações aprendidas pelos descritores. Tal tarefa pode ser definida como: o reconhecimento de um objeto individual como membro de uma classe específica, que contém objetos com características similares. Desse modo, o reconhecimento de objetos é tratado como o ato de rotular uma imagem, dado várias opções de classes para diferentes objetos [6].

Sob esse contexto, busca-se aqui, a realização do estudo e implementação de uma das técnicas de reconhecimento de objetos em imagens, baseando-se no descritor Histograma de Orientações dos Gradientes (*Histogram of Oriented Gradient* - HOG) e nas Máquinas de Vetor Suporte (*Support Vector Machines* - SVM), no intuito de validar um sistema de reconhecimento de objetos.

Este artigo encontra-se dividido da forma que se segue: na Seção II encontra-se a fundamentação teórica para desenvolvimento do projeto, a qual detalha os conceitos básicos para reconhecimento de objetos, como também o descritor e o classificador utilizados. A Seção III traz os materiais e os métodos adotados para implementação do sistema de reconhecimento de objetos. Na sequência, a Seção V apresenta os resultados usados para validação do sistema e suas respectivas discussões. Por fim, a Seção VI apresenta as conclusões desse estudo.

II. Fundamentação Teórica

A. Reconhecimento de Objetos

Geralmente, as atividades de reconhecimento de objetos são realizadas em duas etapas: extração de características e classificação, com duas fases distintas de operação: treinamento e reconhecimento.

A fase de treinamento do algoritmo consiste na busca de padrões para representar os objetos de interesse. Esses padrões são discriminados por um conjunto de características extraídas usando um descritor que pode ser local ou global, ou seja, o descritor atua ao redor de pontos de interesse ou na imagem como todo [1]. As informações extraídas pelos descritores podem conter: aspectos geométricos, estatísticos, informações de cor, profundidade, borda, textura, entre outros [2].

Nessa fase, dado um conjunto de imagens para treino, o descritor é aplicado, em seguida, as características extraídas são submetidas a um algoritmo de aprendizado de máquina (AM), que por meio de algoritmos de agrupamento usando funções matemáticas, buscam separar as características provenientes dos descritores que diferenciam o objeto de interesse dos demais, gerando um modelo dos padrões [2]. Alguns exemplos

desses algoritmos são a Máquina de Vetor Suporte (*Support Machine Vector* - SVM) e os algoritmos genéticos [7].

Para reconhecer um objeto em uma imagem, é realizado um casamento dos padrões que correlaciona o objeto de interesse com os de uma classe específica. Para isto, dado uma imagem, os vetores de características (descritores) são extraídos. A seguir é realizada a classificação, através de uma medida de similaridade entre os descritores da imagem analisada e de todos os modelos de descritores que descrevem os objetos das classes aprendidas. Na Figura 1 é apresentado o esquema geral para um algoritmo de reconhecimento de objetos.

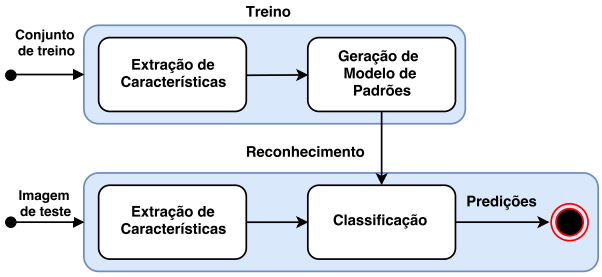


Figura 1: Esquema geral para algoritmo de reconhecimento de objetos.

B. Histograma de Orientações dos Gradientes

O HOG é um descritor de características introduzido por [8]. Esse descritor utiliza o fato de que a aparência de um objeto local e seu formato podem ser caracterizados pela intensidade de gradientes e pelas direções.

O algoritmo de extração de características HOG é realizado de acordo com os seguintes estágios descritos em [9]:

- O primeiro estágio consiste em calcular os gradientes de primeira ordem da imagem nas direções x e y. O cálculo dos gradientes captura contornos, silhuetas e um pouco de textura da imagem. As máscaras espaciais usadas para este cálculo são dadas por: $[-1\,0\,1]$ para o cálculo do gradiente no eixo-x e $[-1\,0\,1]^T$ para o cálculo do gradiente no eixo-y. O gradiente da imagem é obtido utilizando como entrada a imagem em tons de cinza.
- O segundo estágio consiste na criação dos histogramas de orientações dos gradientes, que tem como objetivo produzir uma codificação que seja sensível ao conteúdo local da imagem e ao mesmo tempo resistente a pequenas variações de pose e aparência. Em [8], os autores utilizaram um redimensionamento de tamanho das imagens igual a 64×128 *pixels*, sendo essa dividida em células de tamanho 8×8 *pixels*. Cada um dos *pixels* é usado para criar um histograma de orientações dos gradientes, sendo um histograma por célula. Cada histograma possui 9 divisões (*bins*), igualmente espaçados entre 0 e 180 graus.
- O terceiro estágio consiste consiste na normalização dos blocos, que são formados por um conjunto de células. A normalização introduz um aumento na invariância à iluminação, sombreamento e contraste dos contornos. Em [8], cada bloco é formado por um conjunto de 2×2 células. Antes da formação dos histogramas nas células, uma máscara gaussiana do tamanho do bloco (16×16) é aplicada sobre a magnitude dos gradientes dos *pixels* do bloco. Essa máscara possui desvio padrão σ igual a metade da largura do bloco, ou seja, $\sigma = 8$, e possui o intuito de diminuir o peso dos *pixels* que se encontram na borda do bloco. Os blocos possuem uma sobreposição de 8 *pixels* entre eles.
- Por fim, ocorre a concatenação de todos os blocos presentes na imagem, num único vetor, criando assim o descritor final. O número de características HOG é calculado para as especificações de tamanho da imagem, em que os blocos e células são utilizados como parâmetros. Para uma imagem de tamanho 64×128 *pixels*, com células de tamanho 8×8 *pixels* e blocos de tamanho 2×2 células existem 3780 *bins*, que representa a dimensão do vetor HOG. Pois há 15 blocos na direção horizontal, 7 blocos na direção vertical, cada um formado por 4 células e cada célula possui um histograma com 9 bins, desse modo, têm-se a dimensão do descritor calculada da seguinte forma: $15 \times 7 \times 2 \times 2 \times 9 = 3780$.

Na Figura 2 é ilustrado um diagrama representando os passos de aplicação do descritor HOG acima.

C. Máquinas de Vetor Suporte

As Máquinas de Vetor Suporte são uma técnica de aprendizado de máquina supervisionado. Ou seja, tem-se a figura de um professor externo, o qual apresenta o conhecimento do ambiente por conjuntos de exemplos na forma: entrada e saída desejada. O objetivo é que a representação gerada seja capaz de produzir saídas corretas para novas entradas [10].

As SVMs são classificadores discriminativos que atuam na separação de padrões por meio de hiperplanos. Em outras palavras, dado um conjunto de amostras rotuladas para treino, o algoritmo busca ótimos hiperplanos para categorizar as amostras de teste [11]. Para isto, o algoritmo SVM baseia-se em encontrar o hiperplano que gera a maior distância entre amostras de classes diferentes para o conjunto de treino. Essa distância é denominada de margem.

As SVM podem classificar dados linearmente separáveis ou não linearmente separáveis. No caso linear, dado um conjunto de treino X de vetores de características x_i, com $i = 1, 2, ...N$, que pertencem a uma das duas classes ω_1 ou ω_2 linearmente separáveis, o objetivo é encontrar o hiperplano $g(x) = \omega^T x + \omega_0 = 0$ que classifica corretamente todos os vetores de X. Caso não-linear, uma função não linear é usada para levar o conjunto de vetores a uma dimensão maior onde é possível separá-los por um hiperplano [12]. Na Figura 3 é ilustrado duas classes sendo separadas por dois hiperplanos.

III. Materiais e Métodos

A. Materiais Utilizados

Os experimentos desse trabalho utilizaram-se de três bases de dados, as quais são: *Caltech-101*, *MSRC v1* e o *Stanford Cars Dataset*. Maiores detalhes são encontrados a seguir.

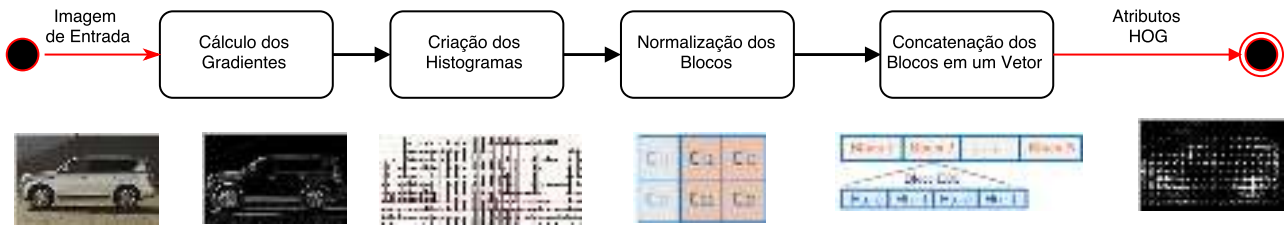


Figura 2: Diagrama de funcionamento do descritor HOG.

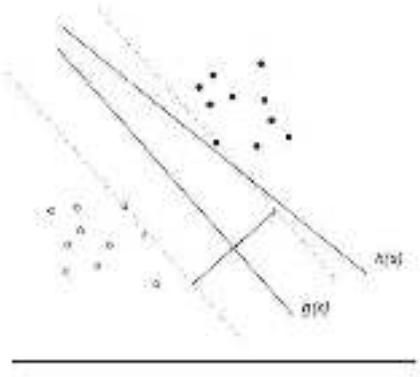


Figura 3: Exemplo de duas classes separáveis por hiperplanos $g(x)$ e $h(x)$ [12].

1) Base de Dados Caltech-101: Essa base de dados consiste em 9146 imagens distribuídas em um conjunto de 101 categorias, cada classe contém aproximadamente entre 40 a 800 amostras e em média suas dimensões são de 300×200 *pixels*. Essas imagens foram coletadas em Setembro de 2003 por Fei-Fei Li, Marco Andreetto e Marc 'Aurelio Ranzato do *California Institute of Technology (Caltech)*, maiores detalhes são encontrados em [13], [14]. Na Figura 4 é apresentado alguns exemplos de objetos dessa base de dados.

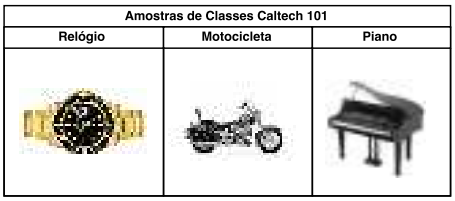


Figura 4: Exemplo de objetos das classes da base de dados *Caltech-101*.

2) Base de Dados MSRC v1: Essa base de dados foi construída pelo *Microsoft Research in Cambridge (MRSC)*, possuindo 240 imagens dispostas em 8 classes com 30 amostras em cada. As imagens desse banco de dados encontram-se disponíveis no site do *The French Institute for Research in Computer Science and Automation (INRIA)*[1]. Na Figura 5 é apresentado alguns exemplos de objetos dessa base de dados.

[1]http://www.labri.fr/perso/yren/database/

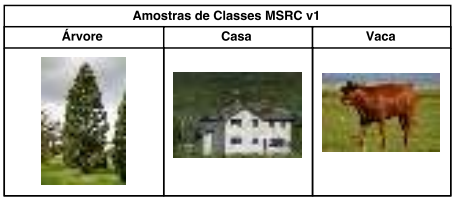


Figura 5: Exemplo de objetos das classes da base de dados *MSCR v1*.

3) Base de Dados Stanford Cars Dataset: Essa base de dados possui 16185 imagens de carros, sendo 8144 imagens para o conjunto treino, 8041 para o conjunto teste, distribuídas em 196 classes. Cada categoria varia entre 24 e 68 amostras, a grande maioria contém em torno de 40 amostras por classe [15]. Tipicamente, as classes definem a marca, o modelo e o ano do carro, por exemplo: 2012 Tesla Model S. Na Figura 6 é ilustrado alguns exemplos de objetos dessa base de dados.

Figura 6: Exemplo de objetos das classes da base de dados *Stanford Cars Dataset* [15].

B. Plataformas Empregadas

1) Biblioteca OpenCV: O OpenCV *(Open Source Computer Vision)* é uma biblioteca de programação, de código aberto, desenvolvida inicialmente pela *Intel Corporation*. Essa biblioteca implementa uma variedade de ferramentas de interpretação de imagens, contendo desde operações simples como um filtro de suavização de ruído, até operações complexas, tais como a análise de movimentos, reconhecimento de padrões e reconstrução em *3D*. O pacote OpenCV está disponível gratuitamente na *Internet*, bem como o manual de referência [16]. A biblioteca está dividida em cinco grupos de funções: processamento de imagens, análise estrutural, análise de movimento e rastreamento de objetos, reconhecimento de padrões e Calibração de câmera e reconstrução *3D* [16].

2) Biblioteca Multi-core LIBLINEAR: A *Multi-core Library for Large Linear Classification (LIBNEAR)* é uma biblioteca de programação paralela de funções, que é usada para resolução de problemas de aprendizagem a partir do algoritmo SVM. Essa biblioteca foi desenvolvida pelo *National Taiwan University (NTU)* por [17], sendo escrita em *C++* com *interface* desenvolvida em *C*. Essa ferramenta implementa apenas modelos lineares do SVM, contendo algoritmos com menor custo computacional devido a sua programação paralela que se utiliza de vários *threads* [18].

IV. Metodologia do Sistema Implementado

O sistema de reconhecimento de objetos implementado extrai características através do descritor HOG e realiza a classificação utilizando o SVM linear, ambos descritos nas Seções II-B e II-C, respectivamente.

O diagrama de blocos do sistema implementado é ilustrado na Figura 7. Como previamente explicado, o sistema possui duas fases distintas de operação: treinamento e reconhecimento. Inicialmente, como pré-processamento, as imagens são redimensionados para um tamanho específico e aplica-se o descritor. Na fase de treinamento, os vetores de características (descritores) são extraídos de cada uma das imagens do subconjunto de treino da base de dados. Esses vetores de características são utilizados para treinar o classificador SVM, gerando os modelos que discriminam cada uma das classes de objetos que o sistema estará apto a reconhecer.

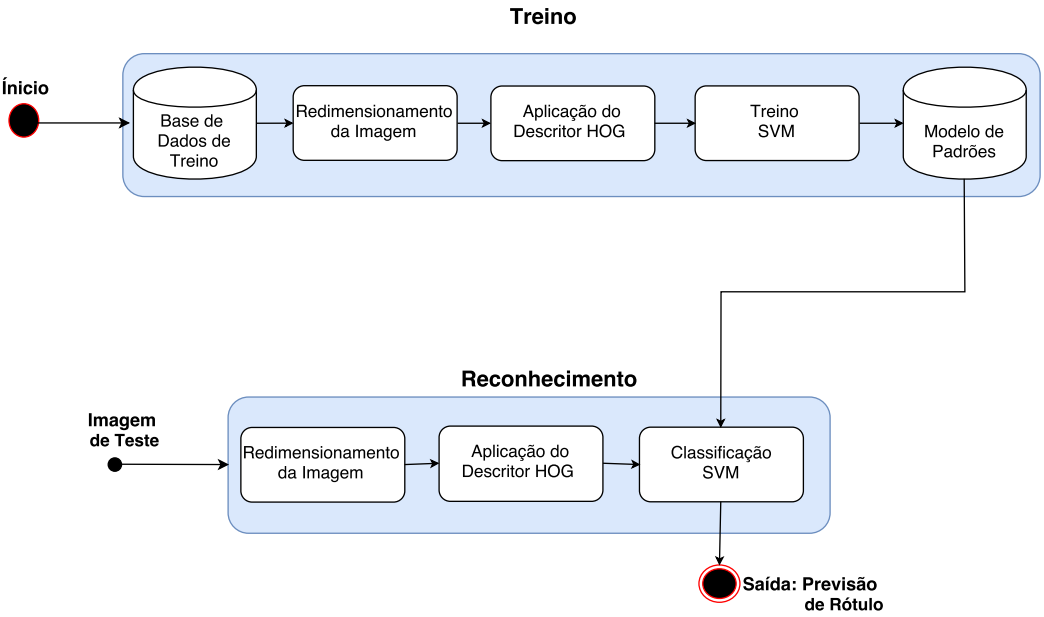
Para a realização dos testes, utilizou-se a fase de reconhecimento do sistema. Para tanto, foi utilizado um novo subconjunto da base de dados denominado conjunto de teste, contendo imagens distintas daquelas utilizadas para compor o subconjunto de treinamento. Para cada imagem no subconjunto de teste, é realizado como pré-processamento, um redimensionamento das imagens para um tamanho específico, em seguida, ocorre a discriminação de suas características pelo descritor HOG. Por fim, o SVM, com o uso do modelo de padrões aprendido, prediz o rótulo da classe a que o objeto corresponde.

Os experimentos nesse trabalho foram estruturados da seguinte forma:

1) Para cada uma das base de dados, conforme descrito anteriormente, as imagens foram divididas em um conjunto de treino e de teste, o de treino para aplicação de descritor HOG e treinamento do SVM, e o de teste para a fase de reconhecimento do classificador.
   - Para o banco de dados *Caltech-101*, foram selecionadas 12 classes de objetos, contendo cada uma 98 imagens distintas, resultando em 1176 objetos, sendo eles divididos nas seguintes classes: avião, motocicleta, face, relógio de pulso, leopardo, bonsai, lateral de carro, barco, candelabro, tartaruga, piano e cérebro. Essa base de dados foi subdividida nos subconjuntos de treino e teste, 80% de suas imagens foram alocadas para o conjunto de treino, e 20% foram alocadas para o conjunto de teste. Logo, o primeiro conjunto ficou com 948 imagens e o segundo com 228 imagens.
   - Para a base de dados *MSRC v1* utilizou-se a mesma metodologia para a criação dos subconjuntos de treino e teste. Nesse caso, a quantidade de imagens nos subconjuntos de treinamento e teste é de 192 e 48 imagens, respectivamente.
   - Para a base de dados *Stanford Cars Dataset* foram utilizadas o as imagens de conjunto de treino e teste como previamente alocadas pelo autor, ou seja, 8144 e 8041, respectivamente. Entretanto, o conjunto de treino não estava muito maior que o de teste e nem balanceado, ou seja, cada classe contendo o mesmo número de amostras. Logo, a solução foi utilizar o processo de *data augmentation*, com o objetivo de a aumentar o número de amostras por classe. A solução foi adotada, pois, uma vez que a quantidade de classes era alta (196 classes) para uma quantidade pequena de amostras por classe (aprox. 40 amostras), o classificador SVM resultaria em baixas taxas de reconhecimento [17], [18].
     Assim, imagens sintéticas foram geradas a partir das originais por meio de espelhamento horizontal, vertical, rotação no sentido horário e anti-horário, aguçamento gaussiano, etc. Esse processo foi realizado com objetivo de se ter 150 imagens por classe, resultando num subconjunto de treino com 29400 imagens.
     Nesse trabalho, com objetivo de comparação do efeito do balanceamento do subconjunto de treino no desempenho do sistema, foram realizados experimentos com os dois subconjuntos de treino (subconjunto de treino não balanceado de [15] e o subconjunto de treino balanceado e aumentado (BA). Em ambos os casos foi utilizado o mesmo subconjunto de teste, originalmente proposto por [15].

2) Após o levantamento das bases de dados, o segundo passo consiste na aplicação do descritor HOG de modo global, ou seja, em toda a imagem. Para esse processo, utilizou-se na etapa de redimensionamentos de imagens, os seguintes tamanho: 64×128 *pixels* (proposto por [8]) e 128×256 *pixels* (sugestão empírica). Os experimentos foram realizados com blocos de tamanho 2×2 células, com células variando de 8×8, 16×16 e 32×32 *pixels*. Essas variações de células ocorreram apenas para os experimentos com as bases de dados da *Caltech-101* e *MSRC v1*, o melhor resultado foi aplicado aos experimentos com a base de dados *Stanford Cars Dataset*, uma vez que esse último apresentava grande esforço computacional para treinamento SVM, mesmo com o *Multi-core* LIBLINEAR.
3) Por fim, a classificação do SVM é realizada para cada base de dados e suas taxas de reconhecimento são avaliadas. Esse resultado é calculado pela métrica acurácia que representa a eficácia do classificador [19], ou seja, a razão do número de acertos sobre o total de objetos a serem reconhecidos. Essa taxa é fornecida pelo próprio *Multi-core* LIBLINEAR após o termino da fase de

Figura 7: Diagrama do funcionamento do sistema de reconhecimento de objetos implementado.



reconhecimento.

Para execução das rotinas e realização dos testes utilizou-se um *laptop* da marca *Acer* com processador *Intel i7–3632QM* de 2.2 GHz, 8 GB de memória *Random-access memory (RAM)* com sistema operacional de 64 *bits* e *Windows 10 Home*. Com exceção do *Multi-core* LIBLINEAR, as rotinas foram implementadas na Linguagem *C++* utilizando-se das funções da biblioteca *OpenCV*, na versão 2.4.13. Além disso, a *Integrated Development Environment (IDE)* utilizada foi o *Visual Studio 2012*.

V. Resultados e Discussões

Esta seção apresenta os resultados experimentais de desempenho dos classificadores SVM utilizando descritores HOG, quando aplicado às três bases de dados distintas. Os resultados estão divididos de acordo com as características de redimensionamento do tamanho da imagem no pré-processamento. São apresentados resultados provenientes da avaliação de desempenho com redimensionamento das imagens para 64×128 e 128×256 *pixels*.

A. Resultados Experimentais com Redimensionamento de Tamanho da Imagem para 64×128 Pixels

Na Tabela I são apresentados os resultados para taxa de reconhecimento do classificador quando submetido as bases de de dados *Caltech-101* e *MSCR v1*, com blocos de 2×2 células e células com 8×8, 16×16 e 32×32 *pixels*. Os melhores resultados de cada experimento encontram-se em negrito, logo, percebe-se que para esse tamanho de imagem, o melhor comportamento do classificador aplicado é com célula de tamanho 16×16 *pixels*.

Na Tabela II são apresentados os valores para taxa de reconhecimento dos experimentos com as bases de treino balanceadas e não balanceadas, conforme descrito na seção III, assim como, os resultados da taxa de reconhecimento para o classificador quando submetido a base dados *Stanford Cars Dataset* com o tamanho de célula de 16×16 *pixels*. Esse tamanho de célula foi escolhido, devido ao fato de ter gerado o melhor comportamento nos experimentos com os outros bancos de dados, para o mesmo redimensionamento da imagem.

Tabela I: Taxas de Reconhecimento para as bases de dados *Caltech-101* e *MSCR v1*.

Tamanho das Imagens 64×128 ***pixels***		
Base de Dados	**Tamanho da Célula em *pixels***	**Taxa de Reconhecimento (%)**
MSCR v1	8×8	58,33
Caltech-101	8×8	71,93
MSCR v1	16×16	**77,08**
Caltech-101	16×16	**83,33**
MSCR v1	32×32	70,83
Caltech-101	32×32	78,07

Tabela II: Taxas de Reconhecimento para a base de dado *Stanford Cars Dataset* com e sem aumento mais balanceamento do conjunto de treino.

Tamanho das Imagens 64×128 ***pixels***		
Base de Dados	**Tamanho da Célula em *pixels***	**Taxa de Reconhecimento (%)**
Stanford Cars Dataset	16×16	**15,58**
Stanford Cars Dataset BA	16×16	**12,85**

B. Resultados Experimentais com Redimensionamento de Tamanho da Imagem para 128×256 Pixels

Na Tabela III são apresentados os resultados para as taxas de reconhecimento do classificador quando submetido as bases de dados *Caltech-101* e *MSCR v1*. Nessa tabela são encontrados os valores quando o bloco é fixado em 2×2 células, com a célula variando de 8×8, 16×16 e 32×32 *pixels*. Para esse redimensionamento, o melhor comportamento do classificador aplicado é com célula de tamanho 32×32 *pixels*.

Tabela III: Taxas de Reconhecimento para as bases de dados *Caltech-101* e *MSCR v1*.

Tamanho das Imagens 128×256 ***pixels***		
Base de Dados	**Tamanho da Célula em *pixels***	**Taxa de Reconhecimento (%)**
MSCR-V1	8×8	60,42
Caltech-101	8×8	62,28
MSCR-V1	16×16	54,17
Caltech-101	16×16	76,32
MSCR-V1	32×32	**70,83**
Caltech-101	32×32	**84,21**

Na Tabela IV são apresentados os resultados para taxa de reconhecimento do classificador quando submetido a base dados *Stanford Cars Dataset* com o tamanho de célula de 32×32 *pixels*. Esse tamanho foi escolhido, devido ao fato, de ter gerado o melhor comportamento nos experimentos com os outros bancos de dados, para o mesmo redimensionamento.

Tabela IV: Taxas de Reconhecimento para a base de dado *Stanford Cars Dataset* com e sem aumento mais balanceamento do conjunto de treino.

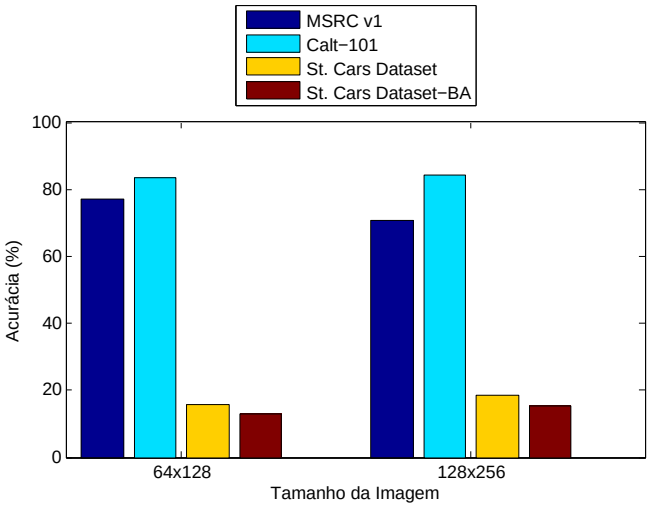
Tamanho das Imagens 128×256 ***pixels***		
Base de Dados	**Tamanho da Célula em *pixels***	**Taxa de Reconhecimento (%)**
Stanford Cars Dataset	16×16	**18,57**
Stanford Cars Dataset BA	16×16	**15,18**

C. Análise Comparativa

Uma análise comparativa dos resultados do classificador é ilustrada na Figura 8. Percebe-se que a taxa de reconhecimento foi maior para a base de dados da *Caltech-101* em ambos os casos de redimensionamento de tamanho da imagem. O desempenho do classificador para a base de dados *MSRC v1* aparece em segundo lugar, em seguida, os experimentos com a base de dados *Stanford Cars Dataset* sem BA aparece em terceiro e, em último lugar, aparece o desempenho do classificador para os testes com a base de dados *Stanford Cars Dataset* com BA.

Com relação aos Experimentos utilizando as bases de dados da *Caltech-101* e *MSRC v1*, os resultados de suas acurácias para o classificador foram iguais ou superiores a 70% em seu melhor comportamento, sendo satisfatórios para o sistema de reconhecimento de objetos com descritor global. Todavia, para os experimentos com a base de dados *Stanford Cars Dataset* não chegaram a 20% em seu melhor comportamento.

Figura 8: Gráfico comparativo do desempenho do classificador quando submetido as três bases e aos dois redimensionamentos de tamanho da imagem.



As justificativas para as baixas taxas de reconhecimento com o banco de dados *Stanford Cars Dataset* giram em torno do uso do descritor HOG e do classificador SVM. Com relação ao descritor HOG, percebe-se que quando a forma dos objetos é muito similar, esses apresentam atributos HOG similares, acarretando em erros de rótulo na classificação. Nesse conjunto de dados, há carros muito parecidos, diferindo apenas a marca deles. Por exemplo, há uma grande variedade de modelos *Pickup* pertencentes a classes diferentes. A Figura 9 apresenta um exemplo desse caso e os atributos HOG extraídos de cada uma.

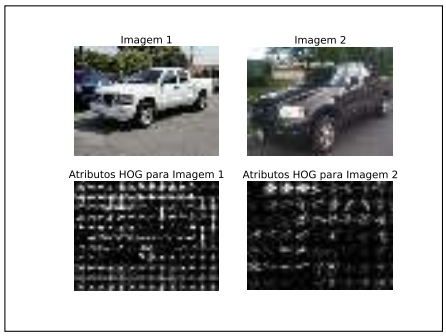


Figura 9: Exemplo da aplicação do descritor HOG em carros do tipo *Pickup* de classes diferente.

Como relação ao SVM, a justificativa gira em torno do fato de que o número de amostras por classe é muito menor que a quantidade total de objetos, acarretando em um fraco desempenho na classificação do SVM [20]. As bases de dados da *Caltech-101* e *MSRC v1* utilizadas possuem uma razão igual a 10, quando comparado o número total de objetos sobre a a média de amostras por classe. Em contraste, para o banco

[1] http://ai.stanford.edu/

de dados *Stanford Cars Dataset* com e sem balanceamento a razão foi de 190 e 200, respectivamente, promovendo um fraco desempenho de classificação.

Um mapa de calor (*Heat map*) é realizado para os resultados com o banco de dados *Stanford Cars Dataset* no intuito de mostrar os erros do classificador com classes de objetos muito similares, no caso dos modelos *Pickup* por exemplo. Esse gráfico representa, em intensidades de cores diferentes, a taxa de reconhecimento em valor absoluto para cada classe, variando de preto para valores próximos a zero, a branco para valores muito altos e intermediando por tons de vermelho. Nele a abcissa representa os rótulos de *ground truth*, enquanto que a *ordenada* representa os rótulos preditos. Um *Heat map* ótimo é aquele que possui a diagonal principal em evidência (branca caso as classes estejam balanceadas) e o resto da matriz em preto, representando valores absolutos de acertos ótimos em todas as classes e nenhum erro de classificação inter-classe.

Para um melhor entendimento, o procedimento adotado foi comparar o *Heat map* das 196 categorias originais com um proveniente do agrupamento de 8 novas categorias, as quais foram: *SUV*, *Sedan*, *Hatchback*, *Conversível*, *Coupe*, *Pickup*, *Táxi* e *Outros*, fornecidos pelas anotações da bases de dados de [15], nessa respectiva ordem quando rotulados de 1 a 8. As Figuras 10 e 11 apresentam os gráficos do mapa de calor para os redimensionamentos adotados nesse trabalho e as Tabelas V e VI mostram uma comparação das taxa de reconhecimento, antes e depois do agrupamento.

Utilizando-se dos mapas de calor é possível perceber em todos os casos, que o agrupamento aumentou os acertos por categoria em torno de mais de 10 vezes e reduziu, consideravelmente, os erros de objetos muito similares. Ao analisar a Figura 10a e 10b, vê-se que o número máximo atingiu 700 acertos em uma classe com o agrupamento, enquanto que sem o agrupamento era de apenas 35 acertos. A classe *SUV*, por exemplo, representada pelo rótulo 1 teve em torno de 700 acertos com o agrupamento e pouco mais de 10 acertos sem agrupamento. Esse resultado mostra que a divisão em grupos, reduz os erros de classificação.

Com maior número de acertos por classe, a taxa de reconhecimento para os experimentos realizados aumentou bastante. Para o redimensionamento de tamanho de imagem igual a 64×128 *pixels*, apresentado na Tabela V, o melhor comportamento variou de 15,58% para 40,47%. Para o redimensionamento de tamanho de imagem igual a 128×256 *pixels*, apresentado na Tabela VI, o melhor comportamento variou de 18,57% para 43,48%.

A Figura 12 ilustra os mapas de calores para os melhores resultados utilizando as bases de dados *MSRC v1* e *Caltech-101*. É possível perceber que a diagonal principal é mais evidente nesses casos quando comparado com os mapas de calor dos experimentos com a base de dados *Starford Cars Dataset*, evidenciando assim maiores acertos e menores erros inter-classe.

VI. Conclusão

Nesse trabalho foi realizado um estudo de uma das técnicas para reconhecimento de objetos em imagens. Para isto, implementou-se um sistema de identificação usando o descritor HOG como global, um classificador SVM e o sistema foi avaliado em três bases de dados distintas.

Tabela V: Taxa de Reconhecimento do classificador aplicado ao banco de dados *Stanford Cars Dataset* com e sem agrupamento.

Tamanho das Imagens 64×128 ***pixels***			
Base de Dados	**Tamanho da Célula em *pixels***	**Taxa de Reconhecimento (n/agrupada) (%)**	**Taxa de Reconhecimento (agrupada) (%)**
Stanford Cars Dataset	16×16	15,58	**40,47**
Stanford Cars Dataset balanceado e aumentado	16×16	12,85	**37,73**

Tabela VI: Taxa de Reconhecimento do classificador aplicado ao banco de dados *Stanford Cars Dataset* com e sem agrupamento.

Tamanho das Imagens 128×256 ***pixels***			
Base de Dados	**Tamanho da Célula em *pixels***	**Taxa de Reconhecimento (n/agrupada) (%)**	**Taxa de Reconhecimento (agrupada) (%)**
Stanford Cars Dataset	16×16	18,57	**43,48**
Stanford Cars Dataset balanceado e aumentado	16×16	15,18	**40,44**

Para os experimentos com as base de dados da *Caltech-101* e *MSRC v1*, o classificador obteve acurácias iguais ou superiores a 70% em seu melhor comportamento, obtendo os valores de 84% e 77%, respectivamente. Esse resultados validam o sistema de reconhecimento de objetos. As falhas ocorridas devem-se, em grande parte, ao fato de que havia grande quantidade de informação de fundo nas imagens, possibilitando assim, falhas de classificação.

Observou-se nos experimentos que o sistema de identificação adotado atua melhor na classificação de objetos de natureza distinta, posto que, a extração de atributos HOG pode apresentar similaridades para objetos de forma semelhante, confundindo o classificador como nos experimentos com o banco de dados *Stanford Cars Dataset*. Notou-se também, que o classificador SVM resulta em baixas taxas de reconhecimento quando o número de amostras por classe é muito menor que a quantidade de objetos. E, mesmo realizando o aumento do conjunto de treino através do processo de *data augmentation*, a razão do número de objetos sobre o de amostras continuou baixa.

Outro ponto a se destacar, é que conjunto de treino com *data augmentation* obteve desempenho inferior ao conjunto de treino original. Uma possível explicação para esse fato é que a introdução de novas imagens inseriu ruídos nas amostras de cada classe, possibilitando assim, a criação de um modelo de padrões ruidoso.

Todavia, verificou-se que após o agrupamento de objetos para o banco de dados *Stanford Cars Dataset*, o problema da similaridade dos atributos HOG em objetos parecidos foi contornado de forma ótima. Por meio dos mapa de calores, foi possível visualizar que os reconhecimento individuais das classes aumentaram em torno de mais de 10 vezes em seu máximo, reduzindo bastante os erros inter-classes. Após o agrupamento, os melhores comportamentos para essa

Figura 10: *Heat map* para os resultados aplicados ao *Stanford Cars Dataset* com tamanho 64×128 *pixels*.

(a) *Heat map* aplicado ao *Stanford Cars Dataset*.

(b) *Heat map* aplicado ao *Stanford Cars Dataset* agrupado.

(c) *Heat map* aplicado ao *Stanford Cars Dataset* BA.

(d) *Heat map* aplicado ao *Stanford Cars Dataset* BA agrupado.

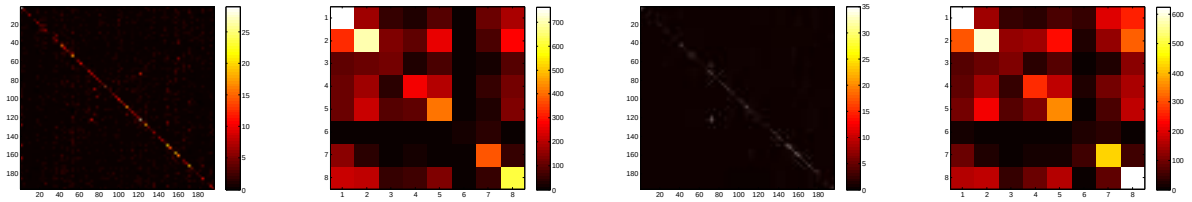


Figura 11: *Heat map* para os resultados aplicados ao *Stanford Cars Dataset* com tamanho 128×256 *pixels*.

(a) *Heat map* aplicado ao *Stanford Cars Dataset*.

(b) *Heat map* aplicado ao *Stanford Cars Dataset* agrupado.

(c) *Heat map* aplicado ao *Stanford Cars Dataset* BA.

(d) *Heat map* aplicado ao *Stanford Cars Dataset* BA agrupado.

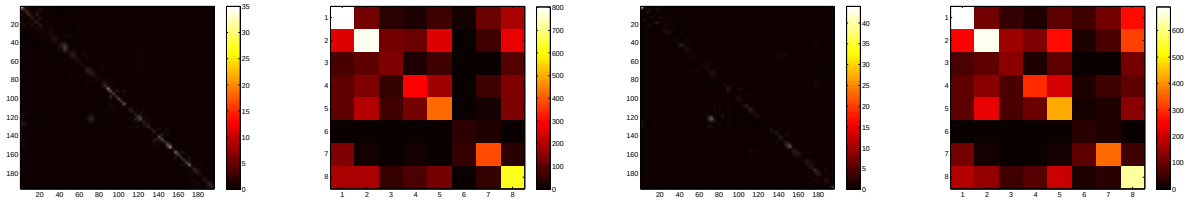
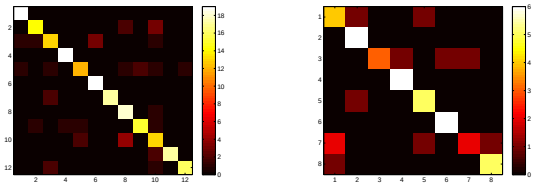


Figura 12: *Heat maps* para os melhores resultados aplicados as bases de dados *Caltech-101* e *MSRC v1*, com 84% e 77% de taxa de reconhecimento, respectivamente.

(a) *Heat map* aplicado ao melhor resultado usando a base dados *Caltech-101*.

(b) *Heat map* aplicado ao melhor resultado usando a base dados *MSRC v1*.



base variaram suas taxas de reconhecimento de 15,58% para 40,47%, para redimensionamento de tamanho da imagem igual a 64×128, e de 18,57% para 43,48%, para redimensionamento de tamanho da imagem igual a 128×256.

Como trabalhos futuros, sugere-se um estudo de outras técnicas de reconhecimento de objetos, utilizando-se de extratores que possibilitem descrever melhor objetos de formas diferentes. Bem como, avaliar a possibilidade de combinar técnicas já existentes e o uso de redes neurais artificiais para extração e/ou classificação.

AGRADECIMENTOS

Os autores agradecem ao Laboratório de Percepção Computacional (LPC) e o Laboratório de Automação, Processamento de Sinais e Imagens (LAPSI), ambos da UFCG, por propiciar a execução desse trabalho.

REFERÊNCIAS

[1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., 2006.

[2] S. Addicam, S. Malik, and P. Tian, *Building Itelligent Systems*. Intel Press., 2012.

[3] R. J. Campbell and P. J. Flynn, "A survey of free-form object representation and recognition techniques," *Computer Vision and Image Understanding*, vol. 81, no. 2, pp. 166–210, 2001.

[4] K. E. A. van de Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders, "Segmentation as selective search for object recognition," in *2011 International Conference on Computer Vision*, Nov 2011, pp. 1879–1886.

[5] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

[6] A. L. Frome, "Learning distance functions for exemplar-based object recognition," Ph.D. dissertation, University of California, Berkley, 2007.

[7] T. M. Mitchell, *Machine Learning (1st Edition)*. McGraw-Hill, Inc., 1997.

[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, June 2005, pp. 886–893.

[9] D. L. Cosmo, "Detecção de pedestres utilizando descritores de orientação do gradiente e auto similaridade de cor," Master's thesis, Universidade Federal do Espírito Santo – UFES, 2014.

[10] A. C. Lorena and A. C. P. L. F. De Carvalho, "Uma introdução às support vector machines," *Revista de Informática e Teória Aplicada - RITA*, vol. 14, no. 2, pp. 43–67, 2007.

[11] O. Documentation, "Introduction to support vector machines — opencv 2.4.13.0 documentation," http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html, 2016, acesso 10 de Julho de 2016.

[12] E. A. Perez, "Descrito de movimento baseador em tensor e histograma de gradientes," Master's thesis, Universidade Federal de Juiz de Fora – UFJF, 2012.

[13] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," in *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04. Conference on*, June 2004, pp. 178–178.

[14] ——, "One-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 594–611, April 2006.

[15] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *4th IEEE Workshop on 3D Representation and Recognition, at ICCV 2013 (3dRR-13)*, 2013.

[16] M. Marengoni and S. Stringhini, "Tutorial: Introdução à visão computacional usando opencv," *Revista de Informática Teórica e Aplicada*, vol. 16, no. 1, pp. 125–160, 2009.

[17] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[18] M. C. Lee, W. L. Chiang, and C. J. Lin, "Fast matrix-vector multiplications for large-scale logistic regression on shared-memory systems," in *Data Mining (ICDM), 2015 IEEE International Conference on*, Nov 2015, pp. 835–840.

[19] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing and Management*, vol. 45, no. 4, pp. 427–437, 2009.

[20] C. wei Hsu, C. chung Chang, and C. jen Lin, "A practical guide to support vector classification," 2010.