

**CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

Mateus Felipe Eisenkraemer

**SISTEMA DE DETECÇÃO DE VEÍCULOS E AVALIAÇÃO DE EIXOS POR MEIO  
DE MÁQUINA DE VETOR DE SUPORTE**

Santa Cruz do Sul

2016

Mateus Felipe Eisenkraemer

**SISTEMA DE DETECÇÃO DE VEÍCULOS E AVALIAÇÃO DE EIXOS POR MEIO  
DE MÁQUINA DE VETOR DE SUPORTE**

Trabalho de conclusão de curso de graduação apresentado ao Departamento de Computação da Universidade de Santa Cruz do Sul como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Área de habilitação: Computação

Orientador: Prof. Dr. Rolf Molz

Santa Cruz do Sul

2016

A capitalist society requires a culture based on images. It needs to furnish vast amounts of entertainment in order to stimulate buying and anesthetize the injuries of class, race, and sex. And it needs to gather unlimited amounts of information, the better to exploit natural resources, increase productivity, keep order, make war, give jobs to bureaucrats. The camera's twin capacities, to subjectivize reality and to objectify it, ideally serve these needs and strengthen them. Cameras define reality in the two ways essential to the workings of an advanced industrial society: as a spectacle (for masses) and as an object of surveillance (for rulers).

(SONTAG, S. *The Image-World*, 1973)

## RESUMO

Este trabalho tem como objetivo elaborar um *software* de detecção e avaliação de eixos de veículos por meio de máquina de vetor de suporte. Com o qual, pretendesse aumentar a agilidade de processos industriais referentes à contagem e avaliação de eixos de veículos que trafegam em um pedágio que foi escolhido como estudo de caso, próximo à cidade de Santa Cruz do Sul. Esta pesquisa é caracterizada por ser de nível acadêmico, exploratória quanto ao levantamento bibliográfico e aos procedimentos técnicos. É também, um estudo de caso, pois foram coletados vídeos dos veículos trafegando no pedágio como forma de amostragem e análise de dados. Para a execução do *software* proposto, foi escolhida a linguagem de programação Python em conjunto com a biblioteca *Open Source Computer Vision Library* (OPENCV). Dentre as técnicas de processamento de imagens utilizadas, destacam-se, *Gaussian Blur*, definição de background, *threshold*, procura por contornos, transformada de *Hough* e histograma de gradientes orientados. O classificador máquina de vetor de suporte foi empregado como forma de validar se os objetos detectados referem-se a eixos de veículos. Testes gerados através do sistema, utilizando filmagens de tráfego viário capturadas pelos próprios autores, indicam que 97% dos carros e 52% dos caminhões e ônibus avaliados tiveram todos os seus eixos contabilizados corretamente.

**Palavras chave:** Transformada de *Hough*, máquina de vetor de suporte, histograma de gradientes orientados, detecção de eixos de veículos.

## **ABSTRACT**

This work aims to develop a software for the detection and evaluation of vehicle axles through the use of a support vector machine classifier. With its use, we intend to increase the agility of industrial processes related to counting and evaluating vehicle axles that travel through a toll that was chosen as a case study, near the city of Santa Cruz do Sul. This research is characterized by being of academic level, exploratory regarding the bibliographic survey and technical procedures. It is also a case study, because it uses video footages as a way of sampling and data analysis. For the implementation of the proposed software, the Python programming language was used in conjunction with the Open Source Computer Vision Library (OPENCV). Among the image processing techniques used, we highlight Gaussian Blur, background definition, threshold, contour detection, Hough transform and histogram of oriented gradients. The classifier support vector machine was employed to validate the detected objects, assuring that they refer to a vehicle axle. Tests using a road traffic footage captured by the authors indicate that 97% of cars and 52% of trucks had their axles correctly accounted by the developed software.

**Keywords:** Hough transform, support vector machines, histogram of oriented gradients, vehicle axl detection.

## LISTA DE FIGURAS

Figura 1 - Etapas do processamento de imagens.....	16
Figura 2 - Fórmula para cálculo de <i>Gaussian Blur</i> . ....	18
Figura 3 - <i>Kernels</i> verticais e horizontais para cálculo do gradiente.....	21
Figura 4 - Exemplo de diferentes linhas que conseguem separar corretamente todos os conjuntos de dados de treinamento. ....	23
Figura 5- Resultados obtidos pelos autores.....	26
Figura 6 - Resultados obtidos pelos autores.....	27
Figura 7 - Resultados obtidos na detecção de capacetes pelos autores.....	28
Figura 8 - Fórmula utilizada pelos autores para avaliação da solução proposta. ....	29
Figura 9 - Resultados obtidos no primeiro experimento pelos autores. ....	30
Figura 10 - Resultados obtidos no segundo experimento pelos autores. ....	30
Figura 11 - Diagrama das técnicas utilizadas no sistema proposto. ....	33
Figura 12 - Local onde foram capturados os vídeos utilizados no estudo.....	34
Figura 13 - Na parte superior, <i>frame</i> em escala cinza, na parte inferior, <i>frame</i> após a aplicação do filtro <i>Gaussian Blur</i> .....	36
Figura 14 - Na parte superior, imagem original, na parte inferior, <i>frame</i> delta da imagem original.....	37
Figura 15 - Na parte superior, <i>frame</i> delta, na parte inferior, <i>threshold</i> do <i>frame</i> delta. ....	38
Figura 16 - Histograma do <i>frame</i> delta.....	39
Figura 17 - Na parte superior, <i>frame threshold</i> , na parte inferior, contornos detectados através da função <i>findContours</i> .....	40
Figura 18 - Na parte superior, contorno detectado, na parte inferior, círculo detectado através da transformada de <i>Hough</i> . ....	42
Figura 19 - Demonstração do incremento do número de eixos do carro com o uso da linha vertical, presente nos dois <i>frames</i> mais a esquerda. ....	44
Figura 20 - Comparação entre imagem de um eixo de caminhão com o seu devido resultado após a aplicação do histograma de gradientes orientados.....	45
Figura 21 - Comparação entre imagem de um eixo de um carro com o seu devido resultado após a aplicação do histograma de gradientes orientados.....	46
Figura 22 - Exemplo de aplicação da técnica <i>sliding window</i> sobre uma região de interesse.....	51
Figura 23 - Exemplo de variação de escala em imagens através de <i>image pyramid</i> . ....	51

<b>Figura 24 - Exemplo de imagem de região de interesse que contempla toda a roda de um caminhão. ....</b>	<b>52</b>
<b>Figura 25 - Exemplo de variação da região de interesse sobre um círculo detectado pela transformada de <i>Hough</i>. ....</b>	<b>53</b>
<b>Figura 26 – Gráfico da quantidade de imagens classificadas como roda nas diferentes variações da região de interesse. ....</b>	<b>55</b>
<b>Figura 27 - Quatro eixos que falharam na detecção de círculos pela transformada de <i>Hough</i>. ....</b>	<b>59</b>
<b>Figura 28 - Eixo onde os contornos detectados foram muito pequenos, não englobando todo o eixo do veículo. ....</b>	<b>60</b>
<b>Figura 29 - Três círculos que não foram validados pelo classificador como sendo eixos válidos de um veículo. ....</b>	<b>61</b>
<b>Figura 30 - Dois círculos de eixos encostados no asfalto que não foram validados pelo classificador como sendo eixos válidos de um veículo.....</b>	<b>62</b>
<b>Figura 31 - Quatro eixos que foram incorretamente avaliados como estando levantados pelo sistema proposto. ....</b>	<b>63</b>
<b>Figura 32 - Imagem da esquerda demonstra o eixo falso positivo contabilizado que interferiu na posição da linha horizontal. As imagens do centro e da direita são os próximos eixos do veículo avaliados como levantados erroneamente. ....</b>	<b>64</b>
<b>Figura 33 - Imagens demonstrando círculos que foram contabilizados como eixos sendo falsos positivos. ....</b>	<b>65</b>
<b>Figura 34 - Imagens demonstrando círculos que foram contabilizados como eixos sendo falsos positivos. ....</b>	<b>65</b>

## LISTA DE TABELAS

Tabela 1 - Tabela comparativa entre os trabalhos selecionados.....	31
Tabela 2 - Resultados obtidos pelo classificador utilizado.....	54
Tabela 3 - Resultados obtidos pelo sistema proposto.....	58
Tabela 4 - Tabela contendo a estrutura utilizada na matriz de confusão para duas classes.....	66
Tabela 5 – Resultados da matriz de confusão para duas classes.....	66



## SUMÁRIO

1	INTRODUÇÃO .....	10
2	JUSTIFICATIVA .....	14
3	OBJETIVOS .....	15
3.1	Objetivo geral.....	15
3.2	Objetivos específicos.....	15
4	FUNDAMENTAÇÃO TEÓRICA.....	16
4.1	Passos fundamentais em processamento de imagens .....	16
4.2	Aquisição de imagens .....	17
4.3	Pré-processamento.....	17
4.3.1	Escala cinza .....	17
4.3.2	<i>Gaussian Blur</i> .....	17
4.4	Segmentação.....	18
4.4.1	<i>Threshold</i> .....	18
4.4.2	Dilatação .....	19
4.4.3	Transformada de <i>Hough</i> .....	19
4.5	Representação e descrição .....	20
4.5.1	Histograma de gradientes orientados .....	20
4.6	Reconhecimento e Interpretação.....	22
4.6.1	Máquina de vetor de suporte .....	22
4.7	Base de conhecimento.....	24
5	TRABALHOS RELACIONADOS .....	25
5.1	Detecção automática de bagas de café em imagens de campo.....	25
5.2	Segmentação, classificação e detecção de motociclistas sem capacete.....	26
5.3	Classificação de Leveduras Utilizando Transformada de <i>Hough</i> e Aprendizagem Supervisionada.....	28
5.4	Tabela comparativa.....	31
5.5	Considerações .....	32
6	SISTEMA PROPOSTO .....	33
6.1	Visão geral.....	33
6.2	Aquisição das imagens .....	34
6.3	Pré-processamento.....	34

6.4	Segmentação.....	36
6.5	Representação e descrição .....	44
6.6	Reconhecimento e interpretação .....	46
7	VALIDAÇÃO DO CLASSIFICADOR .....	49
7.1	Imagens que contém somente rodas .....	49
7.2	Imagens que contém uma região de interesse .....	50
7.3	Imagens que contém uma região de interesse com aplicação de <i>sliding window</i> ...	50
7.4	Imagens que contém uma região de interesse com aplicação de <i>sliding window</i> e <i>image pyramid</i> .....	51
7.5	Imagens que contém uma região de interesse de tamanho variável .....	52
7.6	Avaliação do classificador .....	53
8	RESULTADOS DO SISTEMA PROPOSTO .....	56
8.1	Desempenho .....	56
8.2	Exatidão nas análises de eixo .....	57
8.2.1	Eixos não detectados pela transformada de <i>Hough</i> .....	59
8.2.2	Eixo sem os contornos bem definidos.....	59
8.2.3	Eixos levantados não classificados como rodas .....	60
8.2.4	Eixos encostados no asfalto não classificados como rodas .....	61
8.2.5	Eixos encostados no asfalto detectados como levantados.....	63
8.2.6	Eixos detectados que são falsos positivos.....	64
8.3	Matriz de confusão para duas classes .....	66
9	CONCLUSÃO.....	67
	REFERÊNCIAS .....	69

## 1 INTRODUÇÃO

Foram tiradas e compartilhadas mais fotografias digitais nos últimos anos do que em toda história da fotografia em filme (EVANS; 2015). Mais de dois trilhões de fotos foram compartilhadas em redes sociais e na internet no ano de 2015 (BANDARU; PATIEJUNAS; 2015). Segundo as estimativas feitas em março de 2015, somente no Facebook (2016, <<https://www.facebook.com>>) foram compartilhadas 2 bilhões de imagens diariamente, isto representa 730 bilhões de compartilhamentos de imagens por ano na rede social. Os gigantes da informática atualmente buscam se aperfeiçoar cada vez mais em ferramentas e sistemas de classificação de imagens (DEAN; MONGA; 2015) e mineração de dados, visto o crescimento exponencial desta área nos últimos anos.

Segundo (HAN, 2011) a mineração de dados pode ser vista como a evolução das tecnologias de bancos de dados. Essa técnica já é consolidada no ramo de obtenção, criação e manipulação de dados. Tendo o controle transacional, de integridade e autenticidade como os seus pontos fortes. A obtenção de dados dos mais variados tipos e o seu armazenamento em grandes quantidades é conhecido como "um sistema rico em dados, mas pobre em informações". Portanto, naturalmente, o próximo passo de seu desenvolvimento seria a análise dos dados que o mesmo guarda. Consequentemente, decisões importantes tomadas por gestores muitas vezes não se baseiam em informações e conhecimentos retirados de suas bases de dados, mas sim por intuição ou por processos manuais, simplesmente porque os mesmos não possuem acesso a ferramentas que extraíam conhecimento de suas bases de dados.

Sob o mesmo ponto de vista estão relacionadas às imagens e vídeos digitais, sem a sua devida manipulação e extração de conhecimento, elas têm o seu potencial de auxílio à tomada de decisão descartada. Naturalmente, para que isso não ocorra, foram desenvolvidas técnicas e metodologias que preenchem essa necessidade.

A extração de características locais de uma imagem foi utilizada de forma precursora por Schmid e Mohr (1996) para representar o conteúdo visual em tarefas de reconhecimento de objetos. O cerne desta técnica é a de que as transformações geométricas (mudança de escala e rotação) e fotométricas (alteração no padrão de cores e iluminação) não interfiram nas análises feitas. Sendo que o objetivo é a detecção de pontos ou regiões de interesse que representem algum conteúdo presente na imagem. Essa técnica permite caracterizar precisamente objetos específicos presentes na imagem. Porém, essa representação não é

genérica, pois um mesmo objeto pode estar descrito de diferentes formas dentro de uma mesma imagem.

Uma alternativa é a adição de uma significância às características locais selecionadas. Este método é conhecido como *bag-of-visual-features* (bolsa de características visuais), que é baseada na técnica de *bag-of-words* (bolsa de palavras) (SIVIC; ZISSERMAN, 2003), que já é consolidada na recuperação e representação de informações textuais. Um dos principais ganhos com o uso dessa técnica é a sua representação, que é mais conveniente, pois se trata de um vetor de características de tamanho fixo, ao invés de um conjunto de características locais de tamanho variável.

Em vista disto, após a retirada de características presentes nas imagens digitais, as técnicas de mineração de dados e classificação de objetos são aplicáveis. Com o objetivo de aprender características que se adaptam às bases de dados obtidas, possibilitando fazer previsões e classificação de objetos presentes nas imagens.

Um classificador é uma ferramenta computacional que consegue identificar a categoria de um elemento. Pode ser de dois tipos, supervisionado ou não supervisionado. Ele será do primeiro tipo se depender da existência de um conjunto de dados previamente classificados, que são usados como referência, caso contrário, será do segundo tipo.

A categoria supervisionada baseia-se em duas abordagens: discriminativos ou gerativos. Os classificadores discriminativos se baseiam nos dados de entrada para gerarem uma classe que represente esse conjunto de dados. Alguns dos algoritmos que utilizam essa abordagem são SVM – *Support Vector Machines*, Máquinas de Vetores de suporte (VAPNIK, 1979), *Boosting* (MASON *et al.*, 2000), Redes Neurais Artificiais (ARBIB, 1995) e Regressão Logística (HOSMER; LEMESHOW, 2000). Já os classificadores gerativos aprendem um modelo que classifique os dados de entrada conforme a probabilidade conjunta entre os dados e as classes pré-existentes, fazendo uma classificação estimativa conforme a regra de Bayes. Alguns dos integrantes dessa categoria são GMM – *Gaussian Mixture Model*, Modelo de Mistura Gaussiano (PERMUTER *et al.*, 2003), HMM – *Hidden Markov Model*, Modelo Oculto de Markov (RABINER, 1989), e *Naïve Bayes*, Bayes Ingênuo (NG; JORDAN, 2001).

Técnicas de análise e classificação de objetos em imagens têm sido utilizadas com sucesso em diferentes sistemas de detecção de tráfego e de veículos (ZHOU., CHEUNG., 2016; SIVARAMAN, 2013; ARAGÓN, 2011). Apesar da variação de luminosidade, e de outros fatores que interferem na captura e análise das imagens, segundo MESSELODI *et al.*, 2007, sistemas baseados na análise de imagens oferecem diversas vantagens em comparação a

técnicas mais tradicionais de controle e identificação de tráfego viário. Tendo como vantagem o baixo impacto na infraestrutura das estradas e baixo custo de manutenção. Portanto, um sistema baseado na análise de imagens pode ser adaptado para classificar diferentes veículos ou objetos somente através de suas características visuais.

Tendo em vista as técnicas apresentadas anteriormente, realizou-se um estudo de caso em um pedágio localizado no município de Santa Cruz do sul, onde foi realizada uma filmagem em ângulo de 90° dos carros que passaram nas cancelas para pagamento. O objetivo da filmagem é possibilitar as seguintes análises: fazer a identificação de veículos presentes em *frames* do vídeo; contar o número de eixos presentes em cada veículo; verificar, para o caso de caminhões, se cada eixo está encostado no chão ou levantado. Com o sistema proposto, pretende-se atender a estas demandas de forma eficiente e automatizada, em contraste com a atual forma em que o processo é executado, de forma manual, necessitando intervenção humana.

Para a realização deste trabalho, foram necessários estudos em diferentes áreas. Primeiramente, realizou-se um estudo sobre o referencial teórico utilizado durante o desenvolvimento deste trabalho, que consta no capítulo 4. O capítulo consiste na descrição das etapas dos processos fundamentais em processamento de imagens, onde é abordada a importância e a necessidade de cada uma das etapas para o desenvolvimento de um sistema baseado em visão computacional.

Realizou-se um estudo de trabalhos relacionados, que constam no capítulo 5. Com o intuito de realizar uma análise das técnicas utilizadas atualmente na resolução de problemas similares ao proposto.

Também foram estudadas técnicas específicas que são aplicadas em cada uma das etapas dos processos fundamentais em processamento de imagens. Essas técnicas são abordadas nos subcapítulos da sua etapa correspondente, também contidos no capítulo 5. Entre as técnicas específicas estudadas, estão *Gaussian Blur*, transformada de *Hough*, *threshold*, máquina de vetor de suporte e histograma de gradientes orientados.

O capítulo 6 contém a descrição do sistema proposto. Abordando uma visão geral sobre o mesmo, as tecnologias utilizadas assim como relacionando cada etapa dos processos fundamentais em processamento de imagens com o seu desenvolvimento prático no sistema proposto.

Por fim, o capítulo 7 contém a avaliação dos resultados obtidos pelo classificador utilizado. No capítulo 8 é demonstrada a validação do sistema proposto sobre nova coleta de dados realizada, que é diferente da utilizada para desenvolvimento deste trabalho. O capítulo

9 contém as conclusões e perspectivas futuras para desenvolvimento de outros trabalhos relacionados.

## 2 JUSTIFICATIVA

A combinação do processamento de imagens e a mineração de dados têm se mostrado eficaz na resolução de problemas do mundo real, sendo estes sistemas utilizados em larga escala em diversos campos de estudo, como na aplicação desenvolvida por (SILVA, 2005). Isso ocorre porque estas tecnologias conseguem agregar diversos benefícios graças à agilidade proporcionada pelos meios computacionais. Sendo assim, questiona-se: será possível, por meio do uso de processamento de imagens e classificação de objetos como técnica de mineração de dados, criar um sistema de detecção de veículos, contagem e avaliação de eixos presentes nos *frames* dos vídeos utilizados como estudo de caso, e contribuir nos processos de tomada de decisão com eficiência?

Como justificativa social para o desenvolvimento desse tema, pode-se citar a melhoria e aumento de agilidade em processos industriais, a partir da identificação automática de eixos levantados em vídeo. Além disso, viabiliza também a tomada de decisão precisa por parte de órgãos responsáveis, poupando tempo e recursos.

Como justificativa científica para o desenvolvimento desse tema, tem-se a ampliação da visibilidade dessas técnicas e seus resultados para a área da computação, com isso, aumentando interesse e investimentos para futuras pesquisas, enfatizando o uso das tecnologias de processamento de imagens e classificação de objetos como técnica de mineração de dados nas mais variadas aplicações.

Como justificativa ambiental, pode-se citar a economia de insumos, que podem ser mais bem aproveitados por meio de uma melhor gestão de recursos de produção, gestão essa que pode ser embasada com a implantação do sistema proposto.

### 3 OBJETIVOS

#### 3.1 Objetivo geral

O objetivo principal é desenvolver um sistema capaz de detecção de veículos, contagem e avaliação de eixos por meio de análise de vídeo em um pedágio.

#### 3.2 Objetivos específicos

- Estudar as diferentes técnicas de obtenção de informação de imagens digitais.
- Estudar as diferentes técnicas de análise e classificação objetos em imagens.
- Analisar dentre as técnicas estudadas quais as que melhor se aplicam ao problema de pesquisa.
- Pesquisar trabalhos relacionados ao tema da pesquisa, a fim de obter *insight* sobre como outros autores lidam com os problemas desta área de pesquisa e qual o estado da arte sobre o tema.
- Desenvolver uma aplicação para detectar a entrada de veículos em *frames* de vídeo, contar o número de eixos que o veículo possui e classificar os eixos, em relação a estarem encostados no asfalto ou não.
- Validar a aplicação desenvolvida utilizando grau de acerto e agilidade na identificação.



## 4 FUNDAMENTAÇÃO TEÓRICA

O termo imagem, refere-se à função bidimensional de intensidade da luz  $f(x,y)$ , onde  $x$  e  $y$  denotam as coordenadas espaciais e o valor de  $f$  em qualquer ponto  $(x,y)$  é proporcional ao nível de cinza da imagem naquele ponto (GONZALEZ; WOODS, 2007).

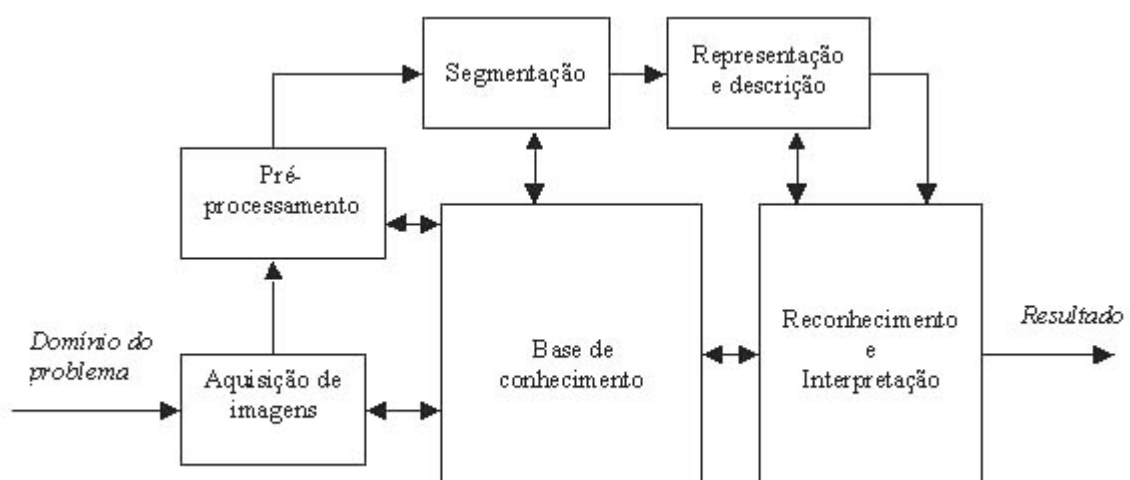
Uma imagem digital é representada por uma matriz cujos índices de linhas e colunas identificam um ponto na imagem, denominado *pixel*, e o correspondente valor do elemento da matriz identifica o nível de cinza naquele ponto.

Sendo assim, esse capítulo aborda conceitos fundamentais necessários para a realização da pesquisa. Conceitos esses que se estendem desde os passos fundamentais em processamento de imagens até a classificação idealizada por máquinas de vetor de suporte.

### 4.1 Passos fundamentais em processamento de imagens

Segundo Gonzalez e Woods, 2007, o processamento de imagens digitais envolve processos onde as entradas e saídas são imagens digitais, envolvendo processos de extração de atributos de imagens e inclusive, o reconhecimento de objetos individuais. Diversas fases compõem o processamento de uma imagem digital, são elas:

Figura 1 - Etapas do processamento de imagens.



Fonte: GONZALEZ e WOODS, 2007.

## 4.2 Aquisição de imagens

O primeiro passo no processo é a aquisição da imagem. Ele necessita de dois elementos, um sensor para aquisição da imagem e um conversor do sinal elétrico analógico produzido pelo sensor para o formato digital. Como exemplo de sensores, têm-se os presentes em câmeras digitais, podendo ser colorida ou monocromática.

## 4.3 Pré-processamento

Após a obtenção de uma imagem digital, o próximo passo trata de pré-processar aquela imagem. O objetivo principal do pré-processamento é melhorar a qualidade da imagem de forma a aumentar as chances para o sucesso dos processos seguintes. Essa etapa envolve principalmente duas categorias principais de métodos, os que atuam no domínio frequência e os que atuam no domínio espacial. Os filtros que atuam no domínio espacial baseiam-se no uso de máscaras, ou de conjunto de *pixels* para modificar a imagem original. Já os filtros de domínio frequência transformam a imagem original para o domínio frequência através da transformada de *Fourier* e aplicam operações, após é feito o processo inverso, transformando a imagem de volta para a forma digital.

### 4.3.1 Escala cinza

Existem diversas representações diferentes para imagens digitais, mas normalmente, são classificadas como coloridas, em níveis de cinza ou binária.

As imagens em nível de cinza possuem valores de *pixel* variando entre 0 (preto) e 255 (branco), todos os valores intermediários constituem a cor cinza. Tendo 256 possibilidades de valor para os *pixels*, é necessário 8 bits (1 byte) para codificar esta informação.

### 4.3.2 *Gaussian Blur*

A técnica de *Gaussian Blur* é largamente utilizada para a redução de ruído em imagens digitais. Após a utilização desta técnica, o efeito encontrado é uma imagem mais suavizada. O filtro gaussiano é um filtro passa-baixa, que reduz a quantidade de *pixels* com alto valor em uma imagem digital. A base de seu funcionamento é a aplicação de convolução na imagem

através de um *kernel*, obtendo um novo valor para cada *pixel* na imagem. O *kernel* pode ser definido como uma matriz de pesos. Convulsionar a imagem com o *kernel* significa que para cada *pixel* da imagem, deve ser colocado o valor do *pixel* atual no centro do *kernel* e seus vizinhos nas suas posições correspondentes então a aplicar o seguinte cálculo.

Figura 2 - Fórmula para cálculo de *Gaussian Blur*.

$$G_0(x, y) = Ae^{-\frac{(x - \mu_x)^2}{2\sigma_x^2} - \frac{(y - \mu_y)^2}{2\sigma_y^2}}$$

Fonte: OpenCV. Disponível em

<[http://docs.opencv.org/2.4/doc/tutorials/imgproc/gaussian\\_median\\_blur\\_bilateral\\_filter/gaussian\\_median\\_blur\\_bilateral\\_filter.html](http://docs.opencv.org/2.4/doc/tutorials/imgproc/gaussian_median_blur_bilateral_filter/gaussian_median_blur_bilateral_filter.html)>. Acesso em: 02/04/2016.

Onde  $\mu$  representa o maior valor de *pixel* correspondente no *kernel*,  $\sigma$  representa a variação, tanto para x quanto para y, “A” representa uma constante de ganho e “e” se refere ao número de Euler.

#### 4.4 Segmentação

O próximo estágio trata da segmentação, que foca em dividir uma imagem de entrada em partes ou objetos menores. Esses objetos referenciam-se a partes da imagem de interesse, ou que contém informação útil para o sistema de processamento de imagens. Da mesma forma, se faz necessário a definição do que é "fundo" da imagem, ou seja, os grupos de *pixels* que podem ser desprezados, que não tem utilidade.

Um procedimento de segmentação robusto auxilia muito na solução bem-sucedida de um problema de visão computacional. É nessa etapa que são definidas as regiões de interesse que serão utilizadas nas análises futuras. Como consequência, quaisquer erros ou problemas presentes nesta etapa serão propagados para as demais, ocasionando resultados negativos.

##### 4.4.1 *Threshold*

A técnica de *Threshold* é o método mais simples de segmentação de imagens, ela é utilizada para criar imagens binárias, definindo o valor de *pixel* somente em 0 (preto) ou 255 (branco), recebendo o valor 0 os *pixels* de *background* e 255 os *pixels* de objetos.

A técnica pode ser aplicada de forma manual ou automática. Através da primeira, é definido um valor base, todos os *pixels* da imagem que tiverem o seu valor inferior ao parâmetro base recebem o valor 0 e os com valor superior, recebem o valor 255. Já na forma automática, o parâmetro de base é definido automaticamente e tem um valor diferente para as diversas regiões da imagem. Essa abordagem é interessante quando se tem diferentes valores de iluminação em diferentes áreas da imagem digital, um simples parâmetro fixo não consegue separar corretamente os objetos do fundo da imagem nas áreas com diferentes valores de iluminação.

#### 4.4.2 Dilatação

Com o objetivo de preencher espaços, a função de dilatação aumenta a área de objetos em relação a áreas de fundo. Para alcançar este objetivo, é estabelecido um *kernel* (matriz) e realizado a iteração sobre todos os *pixels* da imagem, colocando o *pixel* atual da iteração no centro do *kernel*. Então, é feita a substituição do valor de *pixel* atual pelo maior valor de *pixel* presente no *kernel*. O resultado desta técnica é que áreas dos objetos presentes nas imagens digitais tendem a aumentar de tamanho.

#### 4.4.3 Transformada de *Hough*

A transformada de *Hough* é um método para a identificação de formas geométricas em imagens digitais. Ela possibilita a identificação de linhas, círculos, elipses, entre outros objetos. Sua principal vantagem é sua invariância a certas imperfeições de formato que esses objetos possam ter, por exemplo, um círculo pode ser identificado pela técnica mesmo que parte das bordas que o constituem não estejam bem definidas.

Inicialmente a transformada de *Hough* foi modelada para a identificação de linhas em imagens digitais. Porém, a técnica foi estendida para também identificar formatos como elipses e círculos.

Em muitos casos onde se deseja identificar formas diversas como círculos ou elipses, são aplicados filtros de detecção de borda como etapa de pré-processamento. Devido a imperfeições na imagem de origem ou no algoritmo de detecção de borda, há a possibilidade de haver *pixels* faltando nas curvas ou linhas do objeto a ser identificável. Por essas razões, é uma tarefa árdua agrupar as bordas detectadas em linhas, círculos ou elipses. A transformada

de *Hough* combate este problema através do agrupamento de bordas em possíveis objetos a serem detectados. Então é realizado um processo de votação, onde somente os objetos mais semelhantes à forma que se deseja identificar são utilizados como resultado da técnica.

#### **4.5 Representação e descrição**

A representação pode ser constituída por fronteiras de uma região ou por todos os pontos dentro de uma mesma região. No primeiro caso, se concentra nas características da forma, tais como bordas e cantos. Já o segundo, é focado nas propriedades internas das imagens, como textura ou forma.

O processo de descrição, também chamado seleção de características, procura extrair características que resultem em alguma informação quantitativa de interesse ou que sejam básicas para discriminação de objetos (GONZALEZ; WOODS, 2007).

Descritores de imagens descrevem as características visuais do conteúdo de uma imagem digital. Eles descrevem os elementos básicos presentes na imagem, como cor, textura e forma e idealmente, devem ser invariantes a transformações como rotação, luminosidade e ângulo.

##### **4.5.1 Histograma de gradientes orientados**

A técnica de histograma de gradientes orientados (HOG, do inglês *Histogram of Oriented Gradients*) foi apresentada de forma precursora por DALAL e TRIGGS em 2005. O algoritmo calcula o histograma de orientação dos gradientes na imagem. O seu resultado final é um vetor de histogramas extraídos da imagem.

O princípio deste descritor é que a aparência e forma de objetos de uma imagem podem ser descritos através da intensidade dos gradientes ou a direção das bordas, sem um conhecimento prévio da disposição das bordas. São definidas quatro etapas para a geração do descritor: cálculo do gradiente em cada *pixel* da imagem digital, agrupamento dos *pixels* em células, agrupamento das células em blocos e a obtenção final do descritor.

O primeiro passo depende da conversão da imagem para a escala cinza e a sua normalização de iluminação. Após, é feito o cálculo do gradiente, tendo como entrada uma imagem digital, são calculados os gradientes tanto para o eixo x quanto para o eixo y, assim como a sua magnitude e orientação.

O método mais comum para cálculo dos gradientes, ou bordas, é através de um *kernel* de derivação de uma dimensão para um ponto, onde cada *pixel* da imagem é posto no centro do *kernel* pré-definido. Para cada *pixel* na imagem é calculada o gradiente tanto horizontal quanto verticalmente. Ou seja, é realizada a iteração sobre todos os *pixels* da imagem original em escala cinza, e posto cada valor de *pixel* no centro dos *kernels* abaixo para a obtenção das bordas verticais e horizontais da imagem.

Figura 3 - *Kernels* verticais e horizontais para cálculo do gradiente.

$$\begin{bmatrix} -1, 0 -1 \end{bmatrix} \quad \text{e} \quad \begin{bmatrix} -1 \\ 0 \\ -1 \end{bmatrix}$$

Fonte: do autor.

A próxima etapa separa em diferentes regiões, chamadas de células, os *pixels* da imagem digital. Para cada célula é calculado um histograma local de uma dimensão das orientações sobre os *pixels* da célula (SAID et al. 2011).

Cada *pixel* dentro de uma célula é contabilizado para a formação do histograma de orientações daquela célula. O valor que o *pixel* contribui e que é utilizado no histograma é baseado no valor do gradiente do mesmo obtido na etapa anterior. O histograma pode variar entre os canais 0° e 180° ou 0° e 360°.

Após o cálculo dos histogramas, os mesmos são normalizados. O principal benefício da normalização é que efeitos de contraste e iluminação tem menos efeito sobre o histograma das células. Isso é feito pelo acúmulo de histogramas locais em regiões maiores, chamadas de blocos. Os histogramas acumulados são utilizados na normalização das células presentes nos blocos.

Tipicamente, as células geradas estão presentes em mais de um bloco, mas como a sua normalização é feita com base no bloco em que estão presentes, seus valores se tornam diferentes. Então, cada célula pode aparecer mais de uma vez no vetor de saída final do descritor HOG. Apesar de parecer redundância, essa técnica melhora o desempenho do descritor.

A representação mais comum dos blocos é através de retângulos, onde os mesmos atuam como uma *grid*, representados por três parâmetros, o número de células por bloco, o

número de *pixels* por célula e o número de canais utilizados para cálculo do histograma das células.

Após a geração dos histogramas acumulados nos blocos, é gerada a saída do descritor HOG, que consiste em utilizar a disposição das orientações dos vetores de gradientes obtidos para descrever a forma da imagem digital de entrada.

## 4.6 Reconhecimento e Interpretação

O último estágio na figura envolve reconhecimento e interpretação. Reconhecimento é o processo de atribuir um rótulo a um objeto, baseado na informação fornecida pelo seu descritor, ela possui duas etapas principais, o aprendizado e o reconhecimento propriamente dito. A interpretação envolve atribuição de significado a um conjunto de objetos reconhecidos.

O reconhecimento baseia-se na premissa de que objetos similares podem ser agrupados em uma mesma classe ou rótulo. O seu resultado pode ser em percentual, representado o percentual de certeza que um determinado objeto pertence a uma classe específica. Ou também pode resultar em uma nova imagem, com características mais enfatizadas para que um especialista humano possa utilizá-la para a tomada de decisão.

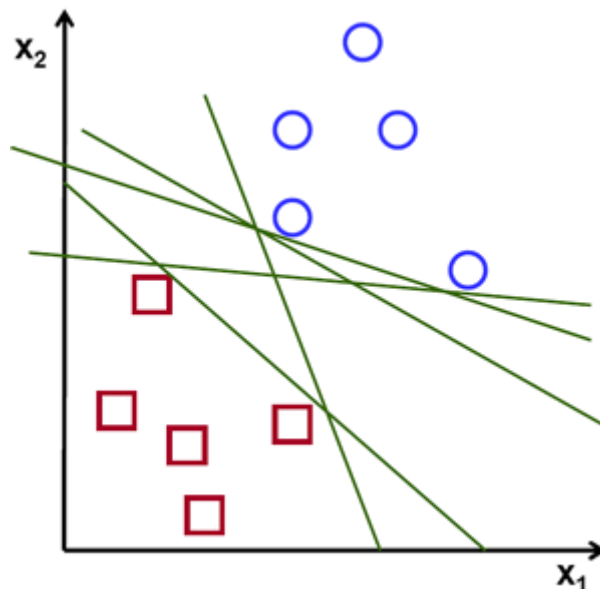
### 4.6.1 Máquina de vetor de suporte

As máquinas de vetor de suporte (*Support vector machines* - SVM) são classificadores de objetos baseados na Teoria de Aprendizado Estático (VAPNIK, 1995). Dentre as principais características, os autores da teoria do aprendizado estático, Vapnik e Chernovenkis (VAPNIK, 1995), destacam as seguintes:

- **Uma boa capacidade de generalização:** os classificadores SVM tem boa capacidade de classificação de dados que não estão contidos no conjunto utilizado em seu treinamento.
- **Robustez em grandes dimensões:** os classificadores SVM conseguem apreender a classificar objetos com grande número de dimensões, sendo imagens digitais um exemplo deste tipo.
- **Teoria bem definida:** SVMs possuem uma base teórica bem definida em campos como matemática e estatística.

Segundo MARSLAND, 2015, outra característica das SVM's é a capacidade de identificar classificadores bons e ruins, mesmo se ambos obtiverem os mesmos resultados de classificação em um conjunto de dados de treinamento. A figura 4 exemplifica esta característica. Apesar de todas as linhas verdes separarem corretamente todos os círculos azuis e quadrados vermelhos, o classificador que se utiliza de máquina de vetor de suporte, irá priorizar a linha que possui a maior distância entre a mesma e o conjunto de dados.

Figura 4 - Exemplo de diferentes linhas que conseguem separar corretamente todos os conjuntos de dados de treinamento.



Fonte: OpenCV. Disponível em

<[http://docs.opencv.org/2.4/doc/tutorials/ml/introduction\\_to\\_svm/introduction\\_to\\_svm.html](http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html)>. Acesso em: 04/10/2016.

A maior região que não contém nenhum dado a ser classificado e que separa a linha que realiza a classificação dos objetos é conhecida como margem. As SVM's buscam o classificador com melhor desempenho de classificação e maior margem dos objetos classificados. Os objetos de uma determinada classe a ser classificada que se encontram mais próximos a linha que realiza a classificação são chamados de vetores de suporte. Esses objetos são os mais importantes dentro do conjunto de dados para o classificador, pois eles determinam a margem sobre uma determinada linha de classificação.



Do trabalho de HEISELE et al, 2001, segue a explicação matemática para a obtenção do classificador com maior margem possível a partir dos dados de entrada. Dado o conjunto de pontos  $x_i \in \mathbb{R}^n$ , com  $i=1,2,\dots,N$ , onde cada ponto  $x_i$  pertence a uma de duas classes distintas. Tais classes são rotuladas  $y_i \in \{-1,1\}$ , a equação para se obter a maior margem possível terá a seguinte forma:

$$f(x) = \sum_{i=1}^l a_i y_i x_i \cdot x + b \quad (1)$$

Os coeficientes  $a_i$  e  $b$  na acima são as soluções para um problema de programação quadrática e a classificação de um novo ponto  $x$  é realizada pela resolução da equação e observação do sinal do resultado, sendo o valor 1 rotulado a uma classe e -1 rotulado a outra.

#### 4.7 Base de conhecimento

O conhecimento sobre o domínio do problema está codificado em um sistema de processamento de imagens na forma de uma base de conhecimento. Esse conhecimento pode ser tão simples quanto o detalhamento de regiões de uma imagem em que se sabe que a informação de interesse pode ser localizada, limitando assim a busca que precisa ser conduzida na procura por aquela informação.

## 5 TRABALHOS RELACIONADOS

Este capítulo consiste na descrição e comparação de um grupo de trabalhos analisados que se utilizam de processamento de imagens e classificação de objetos como solução de um problema específico. Os trabalhos escolhidos possuem um referencial teórico similar ao proposto, por isso, deseja-se analisar as técnicas utilizadas pelos menos assim como seus resultados.

### 5.1 Detecção automática de bagas de café em imagens de campo

A aplicação desenvolvida por SANTOS, 2015, tem como objetivo fazer a detecção automática de bagas de café em imagens de cafeeiros obtidas em campo sob a luz ambiente. Para isso, o autor definiu dois processos distintos, o primeiro sendo a segmentação de áreas da imagem que possivelmente contenham bagas de café. Já a segunda, pretende empregar um classificador para determinar se a região segmentada contém o fruto.

Foram estudadas quatro técnicas de segmentação para a etapa de identificação de regiões que possivelmente contenham bagas de café. Sendo elas, transformada circular de *hough* (YUEN et al, 1989), determinante da *hessiana* (DoH), laplaciano da *gaussiana* (LoG) e diferenças de *gaussianas* (DoG) (TUYTELAARS; MIKOLAJCZYK, 2007). Dentre as técnicas estudadas, LoG e a transformada circular de *Hough* apresentaram os melhores resultados, portanto ambas foram escolhidas pelo autor para serem utilizadas nas etapas seguintes.

Após a obtenção das áreas segmentadas, foram utilizados os descritores de imagens histograma de gradientes orientados (DALAL, TRIGGS; 2005) e DAISY (TOLA et al. 2010) para obter a caracterização da aparência da área previamente segmentada. O resultado dos descritores serviu de entrada para classificadores de objetos. Diversos classificadores foram testados pelo autor, foram escolhidos os dois de melhor desempenho para serem utilizados. Máquinas de Vetores de Suporte (SVM) (VAPNIK, 1979) e *Gradient Boosting* (GB) (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

Os resultados obtidos pelo autor, indicam que utilizando LoG, 97% das bagas de café foram segmentadas corretamente. Já utilizando a transformada circular de *hough*, o resultado obtido foi de 94% de acerto. Portanto, o resultado da segmentação por LoG foi o escolhido para utilização nos descritores e classificadores.

Através da utilização dos dois descritores de imagem e dois classificadores abordados anteriormente, o autor obteve os seguintes resultados.

Figura 5- Resultados obtidos pelos autores

Descritor	Classificador	Precisão	Cobertura
HOG	SVM	0,90	0,63
HOG	<i>Gradient Boosting</i>	0,89	0,65
DAISY	SVM	0,80	0,77
DAISY	<i>Gradient Boosting</i>	0,85	0,73

Fonte: SILVA, 2015.

Como sugestão para trabalhos futuros, o autor sugere a utilização de novos descritores e a combinação de descritores com a redução de sua dimensionalidade.

## 5.2 Segmentação, classificação e detecção de motociclistas sem capacete

Desenvolvido por Silva et al. 2013 o trabalho apresenta um método para a detecção automática de motociclistas sem capacete através de análise de vídeo. Para o objetivo proposto, foram definidas duas etapas principais, a detecção da motocicleta e detecção do uso de capacete.

Foram definidas as etapas de pré-processamento como sendo, transformação da imagem para escala cinza através da distância euclidiana entre as faixas vermelhas, verde e azul no espaço de cores RGB. Além de um filtro de mediana.

Para a detecção de motocicletas, os autores utilizaram técnicas de detecção do plano fundo das imagens utilizadas e segmentação dos objetos em movimento. Foi empregada a técnica *adaptive gaussian mixture model* (ZIVKOVIC, 2004) para a definição de plano fundo.

A segmentação de objetos foi realizada utilizando uma linha, pré-definida pelo usuário da aplicação. Quando um veículo cruzar esta linha, o processo de segmentação de objetos é realizado. Ele inicia com a subtração do plano fundo encontrado anteriormente com o *frame* atual. Sobre o resultado deste processo, é então utilizado o algoritmo de *otsu* (OTSU, 1979) para conversão da imagem para formato binário. Em seguida, é aplicado o algoritmo de *sobel* (SOBEL, 1970) para a detecção de borda e um operador morfológico de erosão.

Na etapa de classificação dos objetos segmentados, podendo ser motocicletas e não motocicletas foi utilizado sobre os objetos segmentados o descritor Padrão Binário Local (*Local Binary Pattern* - LBP). Como classificador, foi utilizado *Support Vector Machine* (SVM) (VAPNIK, 1979).

Após a classificação e identificação das motocicletas, foi utilizada a parte superior das imagens das mesmas como região de interesse para a identificação do capacete. Para a extração de atributos sobre a região de interesse, foi utilizado a transformada circular de *Hough* em conjunto com os descritores LBP e Histograma de Gradientes Orientados (HOG). Para o processo de classificação se o condutor da motocicleta estava de capacete ou não, foram testados três classificadores. São eles, *Naive Bayes*, *Support Vector Machines*, e *Random Forest*.

Para avaliar o desempenho das técnicas e processos descritos acima, os autores adotaram as taxas de sensibilidade (S), especificidade (E), o valor preditivo positivo (VP), valor preditivo negativo (NVP) e a acurácia (A). Os resultados obtidos pelos autores durante a classificação entre motocicletas e não motocicletas estão contidos na imagem abaixo.

Figura 6 - Resultados obtidos pelos autores.

S	E	PPV	NPV	A
0,9502	0,9836	0,9377	0,9870	0,9767

Fonte: SILVA, et al. 2013.

Com a detecção de capacete nas imagens de motocicletas, também foi utilizada a mesma escala de desempenho usado na etapa anterior. Constando os resultados abaixo.

Figura 7 - Resultados obtidos na detecção de capacetes pelos autores.

S	SP	PPV	NPV	A
Naive Bayes				
0.9478	0.7794	0.9603	0.7260	0.9224
Random Forest				
0.9791	0.7353	0.9542	0.8621	0.9423
SVM				
0.9765	0.4117	0.9034	0.7567	0.8913

Fonte: SILVA, et al. 2013.

Os autores concluem que o melhor descritor para a classificação de veículos, dentro dos testados, foi o LBP. Já para a classificação de capacetes nos motoristas, a combinação de HOG, LBP e a transformada circular de *Hough* obtiveram resultados satisfatórios. Em relação aos classificadores testados, observou-se que *Random Forest* obteve melhores resultados.

Como sugestão para trabalhos futuros, os autores indicam que a captura de imagens com maior resolução para utilização no sistema proposto pode gerar melhores resultados, pois muitas imagens não foram utilizadas na fase de detecção de capacete por terem baixa resolução.

### 5.3 Classificação de Leveduras Utilizando Transformada de *Hough* e Aprendizagem Supervisionada

A aplicação proposta por Silva et al, 2012, pretende localizar, classificar e enumerar bactérias de leveduras para automação de testes alcoólicos em imagens microscópicas de fermentação de álcool. O trabalho é dividido em três etapas, detecção das regiões de interesse, que possuem bactérias de levedura, a extração de descritores sobre essas regiões e a sua posterior classificação.

Foram utilizados no pré-processamento, técnicas de saturação, para realce de bordas. Após essa etapa, foi utilizado *Gaussian Blur* para suavizar a imagem, e diminuir o ruído. Posteriormente, a imagem é convertida para escala cinza e é aplicado o filtro *Canny* para encontrar as bordas das leveduras.

Como as leveduras possuem um formato circular, foi utilizado a transformada circular de *Hough* para a sua detecção. Após a aplicação da transformada, é armazenado o raio de

cada círculo detectado, para o cálculo de área das leveduras. Os círculos detectados representam a área de interesse, pois identificam a área em que possíveis leveduras podem estar localizadas.

Para a etapa de extração de descritores das áreas de interesse, foram extraídas informações de forma cor e textura. Para a extração de atributos de forma foram aplicados os algoritmos de K-curvatura e *Shape Descriptors*, já para os atributos de cor foram utilizados os modelos de cor HSB, e para os atributos de textura foi utilizado mapas de Interação.

Após a extração dos descritores, foram testados quatro classificadores supervisionados que tem a tarefa de classificar as leveduras previamente identificadas. São eles, redes neurais artificiais (VAPNIK, 1979), árvores de decisão (UTGOFF 1989), máquinas de vetores de suporte (ARBIB, 1995), e K-vizinhos (GUO et al. 2003). Foram realizados dois experimentos distintos com os classificadores, o primeiro procurando encontrar o classificador que melhor discrimina as diferentes classes de leveduras. Já o segundo, pretende encontrar o classificador que melhor realiza a contagem de leveduras, comparando com a contagem de um perito. A métrica utilizada para comparar os classificadores foi F-Score, que é a média harmônica dos valores de *Recall e Precision* (GOUTTE, GAUSSIÉ, 2005). A fórmula para o cálculo é a mostrada abaixo.

Figura 8 - Fórmula utilizada pelos autores para avaliação da solução proposta.

$$F = 2 * \frac{precision * recall}{precision + recall}$$

Fonte: SILVA et al. 2012.

Os resultados para o primeiro experimento estão abaixo. Onde J.48 são as Árvores de Decisão, IBk os k-vizinhos mais próximos, SMO as Máquinas de Vetores de Suporte e a RNA as Redes Neurais Artificiais. O valor em parênteses é o desvio padrão para cada técnica. Os três tipos diferentes de leveduras classificadas são Brix 3, 6 e 12.

Figura 9 - Resultados obtidos no primeiro experimento pelos autores.

	J.48	IBk	RNA	SMO
Brix 3	0.984(0.010)	0.987(0.009)	0.986(0.010)	0.987(0.010)
Brix 6	0.966(0.020)	0.971(0.017)	0.967(0.019)	0.969(0.018)
Brix 12	0.972(0.012)	0.973(0.012)	0.969(0.011)	0.969(0.011)

Fonte: SILVA et al. 2012.

Para o segundo experimento, os autores utilizaram o classificador k-NN, constando os resultados abaixo.

Figura 10 - Resultados obtidos no segundo experimento pelos autores.

	Hough + ML	Humana
Brix 3 R1	191	190
Brix 3 R2	216	233
Brix 3 R3	257	255
Brix 6 R1	194	197
Brix 6 R2	178	178
Brix 6 R3	73	67
Brix 12 R1	270	259
Brix 12 R2	265	239
Brix 12 R3	239	210

Fonte: SILVA et al. 2012.

Concluindo, os autores indicam que, através dos resultados obtidos, é demonstrada que a contagem automática de leveduras é estatisticamente semelhante a contagem humana. Assim, o método proposto pode ser aplicado na automação do controle de viabilidade na fermentação alcóolica.

## 5.4 Tabela comparativa

Tabela comparativa entre as três ferramentas relacionadas e o sistema proposto pelo autor:

Tabela 1 - Tabela comparativa entre os trabalhos selecionados.

Nome	Pré-processamento	Segmentação	Representação e descrição	Reconhecimento e Interpretação
<b>Detecção automática de bagas de café em imagens de campo</b>	Escala cinza.	Laplaciano da Gaussiana (LoG). Transformada circular de <i>Hough</i> (YUEN et al, 1989).	Histograma de gradientes orientados (DALAL,TRIGGS; 2005). DAISY (TOLA et al. 2010).	Máquinas de Vetores de Suporte (VAPNIK, 1979). <i>Gradient Boosting</i> (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).
<b>Segmentação, classificação e detecção de motociclistas sem capacete</b>	Escala cinza. Filtro mediana. Mistura de Gaussianas (Zivkovic, 2004). Otsu (OTSU, 1979). Sobel (SOBEL, 1970). Erosão.	Transformada circular de <i>Hough</i> (YUEN et al, 1989).	Padrão Binário Local ( <i>Local Binary Pattern</i> - LBP). Histograma de Gradientes Orientados (DALAL,TRIGGS; 2005).	Máquinas de Vetores de Suporte (VAPNIK, 1979). <i>Naive Bayes</i> (NG;JORDAN, 2001). Random Forest.
<b>Classificação de Leveduras Utilizando Transformada de <i>Hough</i> e Aprendizagem Supervisionada</b>	Escala cinza. Saturação. <i>Gaussian Blur</i> . Filtro Canny.	Transformada circular de <i>Hough</i> (YUEN et al, 1989).	K-curvatura. <i>Shape Descriptors</i> .	Redes neurais artificiais (VAPNIK, 1979). Árvores de decisão (UTGOFF 1989). Máquinas de vetores de suporte (ARBIB, 1995) K-vizinhos (GUO et al. 2003).

Fonte: do Autor.



## 5.5 Considerações

Os três trabalhos descritos na sessão anterior foram escolhidas por meio de busca por artigos e dissertações que envolvam a identificação e classificação de objetos circulares através do uso de processamento de imagens e aprendizado de máquina.

Na etapa de pré-processamento, todos os trabalhos analisados utilizaram a escala cinza. Além desse modelo de cor, os trabalhos relacionados também utilizaram outras técnicas conforme a sua necessidade, utilizando principalmente filtros que realçam bordas nas imagens, a fim de melhorar as detecções futuras.

Nota-se, que todas as propostas utilizam a Transformada circular de *Hough* (YUEN et al, 1989) na etapa de segmentação dos objetos, o que a destaca como um bom candidato para a identificação e segmentação de objetos circulares em imagens digitais.

Nas etapas de representação e descrição e reconhecimento e interpretação, se observa a utilização de várias técnicas distintas, o que demonstra a necessidade de utilização e teste de diversas técnicas para que se encontre a que melhor se adequa ao problema do sistema proposto.

O presente trabalho utiliza na etapa de pré-processamento dos *frames* a escala cinza e o filtro *Gaussian Blur*. Na etapa de segmentação foi utilizado a transformada circular de *Hough* (YUEN et al, 1989), já na etapa de representação e descrição foi utilizado o descritor de imagens histograma de gradientes orientados e como reconhecimento e interpretação foi utilizado o classificador máquina de vetor de suporte (VAPNIK, 1979).

Em comparação aos demais trabalhos mencionados, o presente trabalho caracterizasse por utilizar um conjunto reduzido de técnicas no pré-processamento e por utilizar somente um descritor e classificador de imagens.

## 6 SISTEMA PROPOSTO

Este capítulo descreve a arquitetura proposta para o sistema de detecção de veículos e avaliação de eixos por meio do uso de máquina de vetor de suporte. Nas sessões a seguir, serão introduzidos como os aspectos teóricos demonstrados anteriormente foram desenvolvidos no sistema proposto.

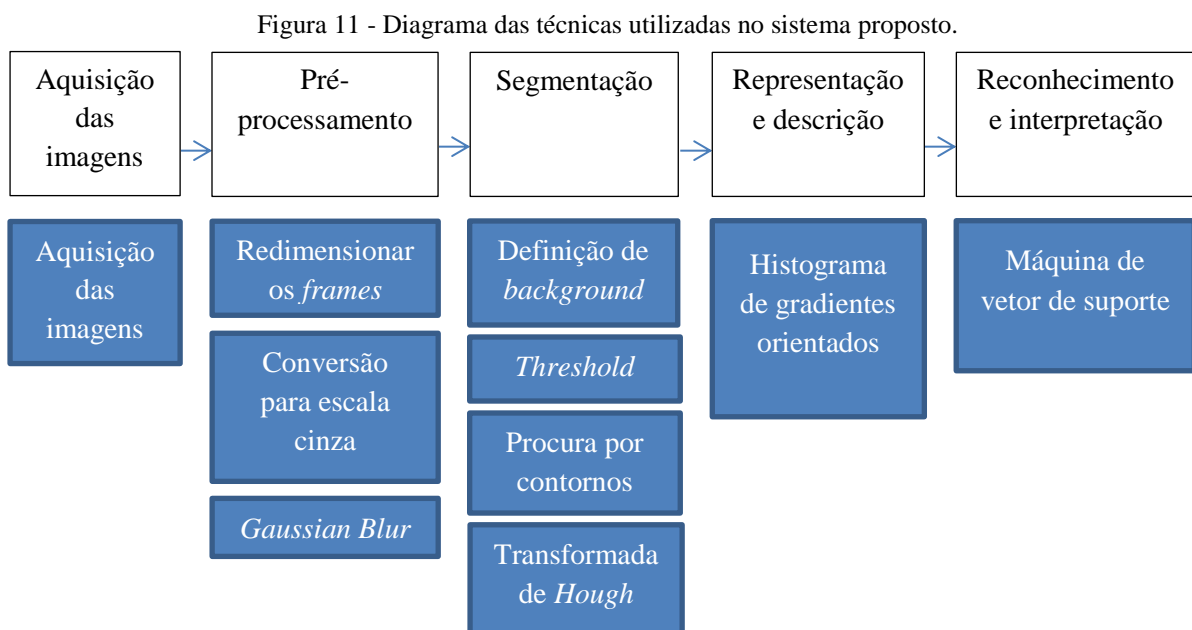
### 6.1 Visão geral

A aplicação proposta possui como entrada um vídeo, gravado previamente no pedágio utilizado como estudo de caso, com o objetivo final de detectar veículos, realizar a contagem de seus eixos e também avaliar se o mesmo está em contato com a rodovia.

A aplicação foi desenvolvida na linguagem Python. O processamento dos *frames* foi realizado com a biblioteca *Open Source Computer Vision Library* (OpenCV, 2016).

O diagrama presente na figura 11 demonstra as principais técnicas utilizadas no sistema proposto em relação aos passos fundamentais em processamento de imagens descritos no capítulo 4.1.

Os próximos capítulos abordam mais detalhadamente as técnicas utilizadas em cada uma das cinco etapas definidas. Seguindo a ordem, aquisição das imagens, pré-processamento, segmentação, representação e descrição e reconhecimento e interpretação.



Fonte: do autor.

## 6.2 Aquisição das imagens

No dia 05/04/2016, com a utilização de uma filmadora própria do autor, foram realizadas filmagens no pedágio utilizado como estudo de caso. As filmagens realizadas foram feitas após a cancela de pagamento por parte dos veículos e feitas em um ângulo de 90°, com o intuito de obter uma visão clara sobre os eixos dos veículos que trafegavam na via. A seguir consta a imagem de um dos locais onde foi realizado filmagens dos veículos.

Figura 12 - Local onde foram capturados os vídeos utilizados no estudo.



Fonte: do autor.

## 6.3 Pré-processamento

A primeira técnica utilizada foi redimensionar o tamanho dos *frames* do vídeo para terem 500 *pixels* de largura, este valor foi definido através dos testes da aplicação e resultou em bons resultados, a altura do *frame* é calculada a partir da proporção original de largura e altura que o *frame* tinha, com o intuito de manter as proporções originais. Esta etapa auxilia no desempenho das análises que irão ser realizadas posteriormente, redimensionando o *frame*, não é necessário processar todos os *pixels* presentes na sua constituição original, mas ainda assim, mantemos a estrutura e contexto das informações presentes neles.

Como a detecção não está relacionada diretamente com a cor do veículo nem com a cor do eixo foi realizada a conversão do *frame* do vídeo para escala cinza. Para isso, foi utilizada a função `cvtColor` presente na biblioteca `OpenCv`, que recebe como parâmetro uma imagem colorida e a transforma automaticamente para escala cinza. Isso permite diminuir a quantidade de informação que deve ser processada por *pixel*, visto que a imagem colorida contém os valores RGB para cada *pixel*, e a imagem cinza somente contém um valor, compreendido entre 0 e 255.

Para diminuir o ruído presente nas imagens, foi aplicada a técnica *Gaussian Blur*. Após a utilização desta técnica, o efeito encontrado é uma imagem mais suavizada. A biblioteca `OpenCV` disponibiliza a função *GaussianBlur*, que recebe como entrada a imagem que irá ser convertida, o tamanho do *kernel*, que deve ser quadrado, e o valor de variação, tanto para x quanto para y.

O *kernel* utilizado tem o tamanho de 21 *pixels* de largura e altura e foi definido através dos testes realizados na aplicação. Quanto maior o tamanho do *kernel* mais a imagem original irá ficar suavizada. Portanto, como objetivo é somente remover ruído presente na imagem original, não é aconselhável utilizar um *kernel* muito grande, pois se pode acabar perdendo informações importantes na imagem.

O valor de variação utilizado tanto para x quanto para y foi 0. Isso porque, quando a biblioteca `OpenCV` recebe estes dois valores, ela calcula o melhor valor de variação para x e y automaticamente sobre os *pixels* da imagem que estão contidos dentro do *kernel*.

Na figura 13 se pode verificar o resultado obtido. Na imagem superior está à imagem original e na imagem inferior a imagem filtrada.

Figura 13 - Na parte superior, *frame* em escala cinza, na parte inferior, *frame* após a aplicação do filtro *Gaussian Blur*.



Fonte: do autor.

## 6.4 Segmentação

A detecção de entrada e saída de veículos envolve diretamente a separação entre o *background* e os demais objetos. Para isso, é necessário ter uma visão clara da estrutura do *background*. Porém, essa estrutura não é fixa, ela muda através dos diversos *frames* do vídeo, como exemplo existe variação de iluminação e novos objetos que são introduzidos no *background*. Para que seja possível a adaptabilidade da detecção, devem-se considerar essas mudanças.

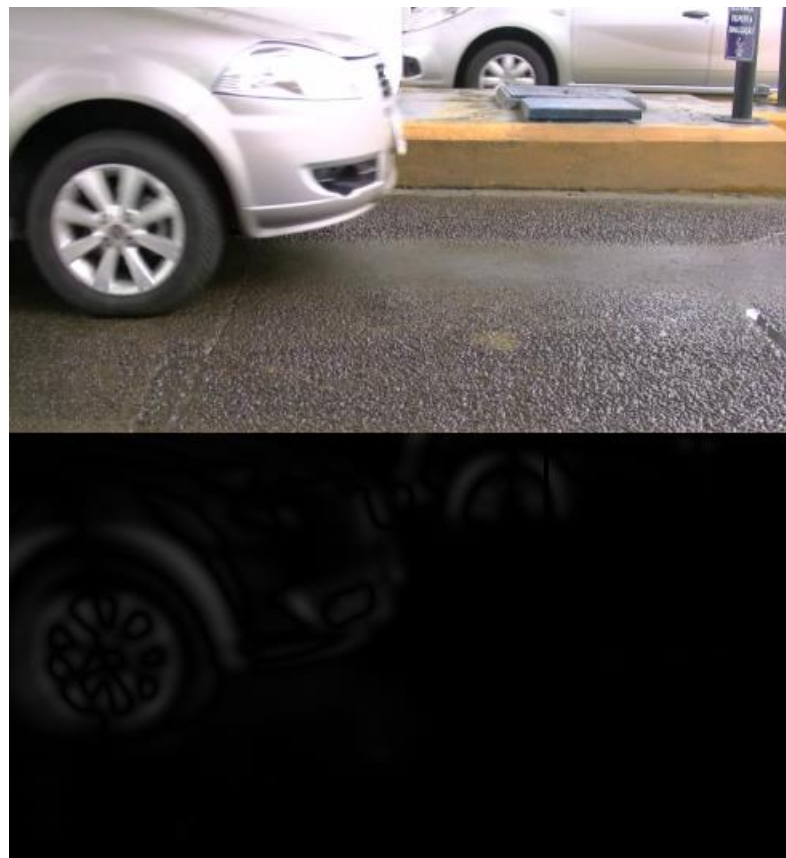
Foi utilizada a função *accumulateWeighted* presente na biblioteca OpenCV. Essa função recebe duas imagens como parâmetro, onde se utilizou o *frame* atual do vídeo e o *frame* anterior, além de um valor de peso correspondente a influência do *frame* atual no resultado final. O valor de peso utilizado para o *frame* atual foi de 0.7, o que significa que ele tem peso de 70% sobre o resultado, o *frame* anterior tem 30% de influência. A saída da função é a média das duas imagens levando em consideração o peso definido para a primeira

e a segunda imagem. É através desta média que se definiu o que é *background*. Desta maneira, a definição não é fixa, e se adapta a diferenças de luminosidade e demais objetos que possam ser introduzidos nos *frames*.

Após a obtenção do *background*, é utilizada a função *absdiff*, também presente na biblioteca OpenCV. A função recebe duas imagens e calcula a diferença absoluta entre as duas. Para isso, foi repassado à imagem de *background* que foi obtida anteriormente e o *frame* atual do vídeo, o resultado é uma nova imagem, chamada de *frame* delta. O *frame* delta contém somente os objetos que tiveram uma variação de posição dentro do *frame*, e que, portanto, devem ser analisados, pois provavelmente indicam o movimento de um veículo.

O *frame* delta, por ser baseado na subtração entre o *background* e o *frame* atual, não contém valores de *pixel* muito alto, dificultando sua análise e usabilidade. Na figura 14 pode ser visualizada esta questão. Na imagem superior está à imagem original e na inferior o *frame* delta da mesma.

Figura 14 - Na parte superior, imagem original, na parte inferior, *frame* delta da imagem original.



Fonte: do autor.

Com o intuito de obter uma visão mais clara do *frame* delta, é aplicado a função *threshold*, presente na biblioteca OpenCV. Para isto, é passada como parâmetro a imagem

original, e um valor limite, onde qualquer *pixel* da imagem que tenha valor menor do que o valor limite estabelecido receberá como novo valor 0 (preto), caso contrário receberá o valor 255 (branco). Desta maneira, é amplificada a diferença entre os objetos que se quer analisar e o *background*, que não será analisado.

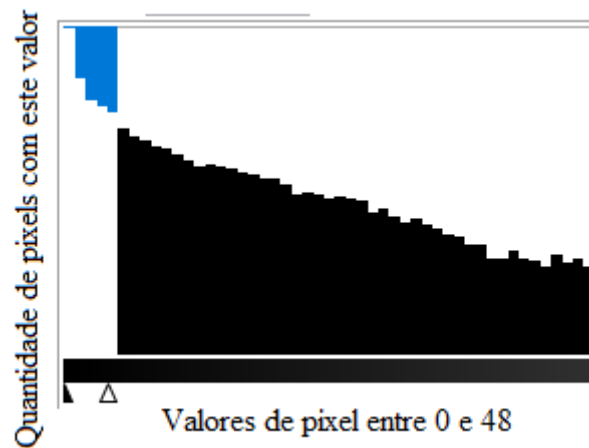
O valor de *pixel* limite utilizado foi 5. Isso porque o *frame* delta contém valores baixos para os seus *pixels*, como pode ser visualizado na figura 15. Os *pixels* do *frame* delta tem pouca luminosidade, se o valor limite fosse mais alto, o retorno da função *threshold* seria uma imagem toda preta, sem detecção de nenhum objeto.

Figura 15 - Na parte superior, *frame* delta, na parte inferior, *threshold* do *frame* delta.



Fonte: do autor.

A imagem 16 contém o histograma do *frame* delta, que foi demonstrado na parte superior da imagem 15. A área demarcada pelas duas setas inferiores demonstra a área da imagem que tem seus valores de *pixel* entre 0 e 4, essa área corresponde a 82.9% da totalidade de *pixels* da imagem. A área que não está contida entre as duas setas varia do valor de *pixel* 5 a 48 e corresponde a 16.9% da totalidade de *pixels* da imagem, é essa área que, após a aplicação da função *threshold* irá ficar com os seus valores de *pixel* em 255(branco). Foram realizados testes para a definição do valor limite utilizado na função *threshold*, sendo que o valor 5 obteve os melhores resultados.

Figura 16 - Histograma do *frame* delta.

Fonte: do autor.

Após o processamento descrito anteriormente, e com a intenção de preencher espaços dos objetos detectados, onde a sua estrutura não foi bem capturada pelos processos anteriores, é utilizada a função *dilate*, presente na biblioteca OpenCV. O efeito dessa técnica é que regiões mais claras da imagem tendem a aumentar. Como anteriormente foi realizada a operação de *threshold*, o efeito da dilatação faz a área dos objetos aumentar, preenchendo eventuais falhas no seu contorno.

Concluindo a etapa de detecção de veículos, deve-se segmentar as partes da imagem que contém objetos identificáveis. Para isto, foi utilizada a função *findContours*, presente na biblioteca OpenCV, utilizando como parâmetros a imagem que se deseja identificar os contornos, o modo de obtenção dos contornos e o método de compressão dos pontos presentes nos contornos. O modo utilizado foi *CV\_RETR\_EXTERNAL*, onde a biblioteca só retorna os contornos mais externos, por exemplo, se houver a detecção de dois contornos, o primeiro abrangendo uma área da imagem e o segundo estando contido dentro do primeiro, o segundo não iria ser identificado como um contorno válido, pois somente os contornos mais externos são detectados. Os outros modos disponibilizados pela biblioteca são:

- *CV\_RETR\_LIST*, onde é retornada uma lista de todos os contornos detectados, não levando em consideração a hierarquia entre eles.
- *CV\_RETR\_CCOMP*, que retorna uma lista de contornos organizando-os em hierarquias de contornos internos e externos.
- *CV\_RETR\_TREE*, que retorna uma lista completa da hierarquia de contornos, possibilitando haver vários níveis distintos de contornos que se englobam.

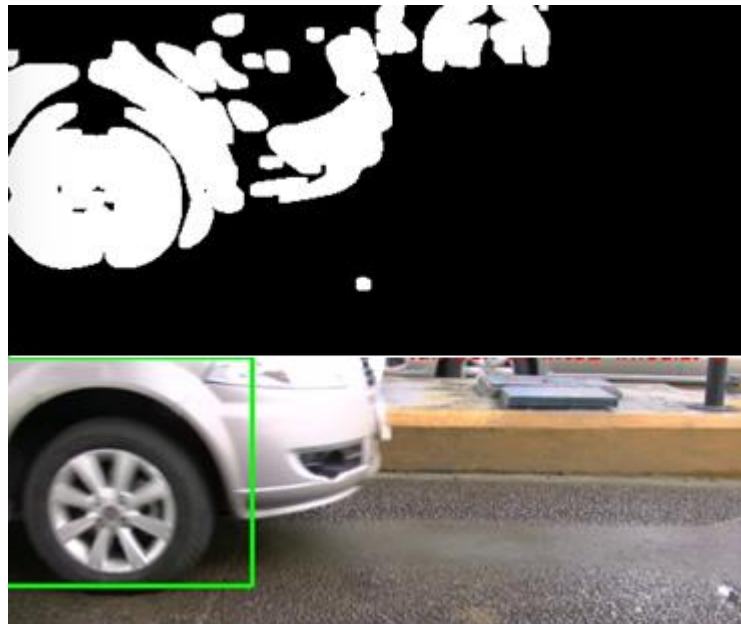


O parâmetro referente ao método de aproximação dos contornos foi definido como sendo `CV_CHAIN_APPROX_SIMPLE`, onde são retornados menos pontos dos contornos detectados, como exemplo, o contorno de um retângulo só retornará os quatro pontos de suas arestas. Os outros métodos de aproximação disponíveis na biblioteca OpenCV são:

- `CV_CHAIN_APPROX_NONE`, são guardados todos os pontos referentes a um determinado contorno.
- `CV_CHAIN_APPROX_TC89_L1`, utilizado para retornar somente os pontos mais importantes em um contorno de curvaturas.
- `CV_CHAIN_APPROX_TC89_KCOS`, versão diferente do método `CV_CHAIN_APPROX_TC89_L1`, também utilizado para retornar os pontos mais importantes em curvaturas.

A detecção dos contornos auxilia na segmentação dos objetos encontrados no *frame* original, delimitando as partes da imagem que serão analisadas nas etapas seguintes. Na figura 17 se pode verificar na imagem superior o *frame* após a aplicação de *threshold* e na imagem inferior a imagem original demarcada com os contornos detectados, contidos no quadrado verde.

Figura 17 - Na parte superior, *frame threshold*, na parte inferior, contornos detectados através da função *findContours*.



Fonte: do autor.

A partir da segmentação da área dos objetos detectados nos *frames* do vídeo, os próximos desafios encontrados foram à detecção de rodas nas áreas segmentadas, a contagem

de rodas presentes em cada veículo e a avaliação do eixo, detectando se o mesmo está encostando no asfalto ou não. Foi utilizada a transformada de *Hough* para a detecção de círculos no *frame*, uma linha vertical para a contagem de rodas por veículo, e utilizado o centro dos círculos detectados pela transformada de *Hough* para a avaliação de eixo. Nos próximos parágrafos, serão detalhadas cada uma dessas operações.

Foi utilizado a função *HoughCircles* presente na biblioteca OpenCV para a detecção de círculos. Os parâmetros necessários para a sua utilização são:

- A imagem que se deseja identificar os círculos.
- O método de detecção, sendo *cv2.HOUGH\_GRADIENT* o único método disponível na biblioteca OpenCV, que é baseado no artigo criado por Yuen et al, 1989.
- Relação inversa entre a resolução do acumulador e da imagem original. Foi utilizado o valor 1, o que significa que o acumulador terá a mesma resolução da imagem original.
- Distância mínima entre o centro dos círculos detectados. Foi utilizado o valor 200, pois não se deseja obter círculos muito próximos uns dos outros.
- Valor do gradiente utilizado na detecção de borda presente no método proposto por Yuen et al, 1989. Foi utilizado o valor 35, que resultou em boas detecções.
- Valor de *threshold* para o controle de quantidade de círculos detectados. Quanto maior o valor de *threshold*, menos círculos irão ser detectados. Foi utilizado o valor 35, que resultou em boas detecções.
- Valor mínimo do raio dos círculos detectados. Foi utilizado o valor 30, pois não se quer detectar círculos muito pequenos,
- Valor máximo do raio dos círculos detectados. Foi utilizado o valor 0, pois não há restrição máxima para o tamanho do círculo detectado.

Na figura 18 se pode verificar a aplicação da função *HoughCircles*. A imagem superior contém o *frame* original com os contornos detectados demarcados pelo quadrado verde. A função *HoughCircles* é aplicada nos *pixels* contidos dentro da área demarcada pelo quadrado verde. A imagem inferior demonstra o círculo detectado pela função além de demarcar o centro do mesmo.

Figura 18 - Na parte superior, contorno detectado, na parte inferior, círculo detectado através da transformada de *Hough*.



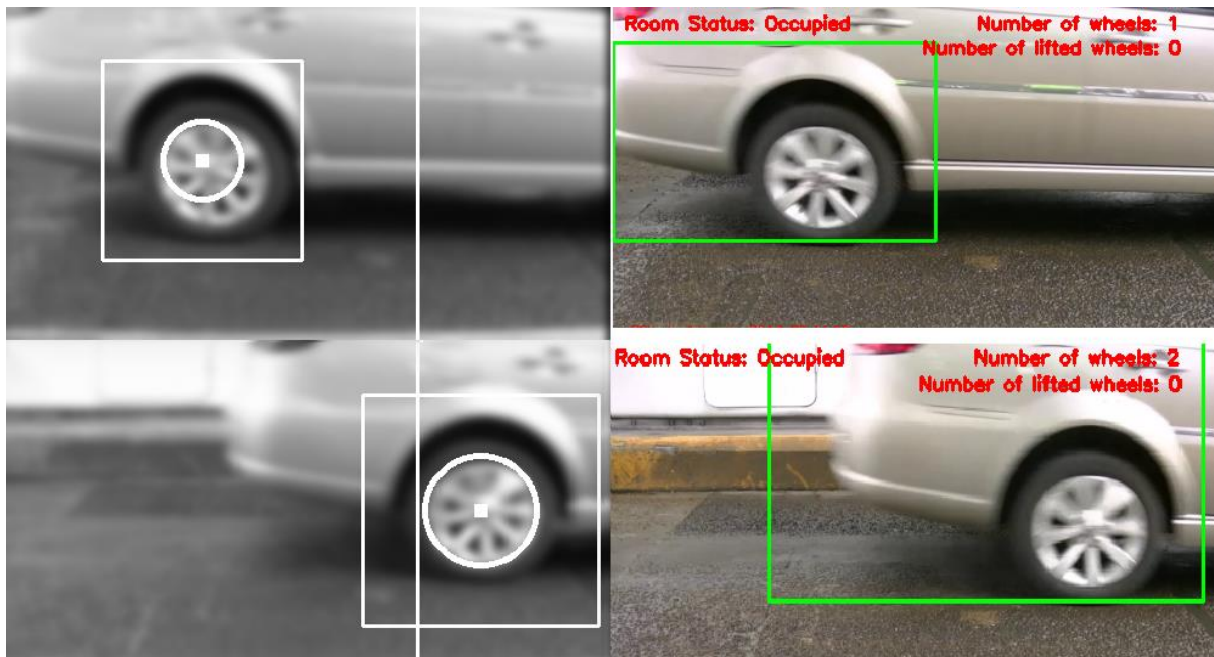
Fonte: do autor.

Devido à quantidade de parâmetros necessários no método, foram realizadas várias iterações de ajuste dos mesmos para que se chegue a um resultado aceitável. Porém, como o objetivo é a detecção de eixos e a transformada de *Hough* faz a identificação de círculos, ocorreram casos onde foram detectados círculos que não eram eixos de um veículo. Assim como demonstrado na parte inferior da imagem 18, onde a transformada de *Hough* detectou a calota do veículo como um círculo válido. A alternativa utilizada para a não contabilização de círculos que não representavam o pneu de um veículo será tratada no próximo capítulo, onde através da utilização de descritores de imagens digitais, foi realizado o treinamento de uma máquina de vetor de suporte para validação do reconhecimento dos círculos detectados, verificando se o mesmo representa um eixo de veículo ou não.

Os círculos detectados pela transformada de *Hough* mudam constantemente de posição nos diferentes *frames* do vídeo, pois os veículos que atravessam o vídeo estão em movimento. Porém, os eixos de cada veículo que está presente no vídeo devem ser contabilizados de forma unitária. Para que o mesmo eixo não seja contabilizado múltiplas vezes, foi utilizada uma linha vertical que separa o *frame* do vídeo em duas partes distintas. Foi salvo em memória o valor no eixo X do centro de todos os círculos detectados no *frame* atual e no *frame* anterior, para que seja possível verificar o movimento dentro do *frame* que cada círculo realizou. Portanto, para que um círculo seja contabilizado como roda do veículo, o mesmo deve estar localizado no *frame* atual com a sua posição de centro em relação ao eixo X posteriormente ao da linha criada para separar o *frame* do vídeo, também, no *frame* anterior, o círculo deve estar em uma posição anterior à linha criada. Desta maneira, é possível restringir que somente serão contabilizados círculos que contém um gradativo movimento dentro dos *frames*, garantindo que não serão contabilizados círculos esporádicos detectados.

Na figura 19 se pode verificar o funcionamento da contagem de eixos por veículo. A imagem superior esquerda demonstra que o círculo detectado está anterior a linha vertical utilizada. Já a imagem superior direita demonstra que a contagem de eixos para este veículo está com o valor um, através do parâmetro *Number of Wheels* no canto superior direito. Já a imagem inferior esquerda demonstra que o eixo foi detectado após a linha vertical, portanto, como demonstrado na imagem inferior direita, o parâmetro *Number of Wheels* foi incrementado para o valor dois.

Figura 19 - Demonstração do incremento do número de eixos do carro com o uso da linha vertical, presente nos dois *frames* mais a esquerda.



Fonte: do autor.

Para a determinação se cada uma das rodas detectadas está encostada no asfalto, foi utilizado o centroide dos eixos já identificados em relação ao eixo  $y$ . As rodas levantadas, que não encostam o seu pneu no asfalto enquanto o veículo trafega na rodovia, tem o seu centro mais elevado em relação às demais, o que possibilita, através da comparação com a média dos centros previamente calculada, fazer esta identificação. Somente foi acumulado na média o centro dos eixos não levantados para cálculo, isso porque os eixos levantados aumentariam o valor da média, podendo ocasionar em muitos eixos levantados detectados como falso positivo. A regra definida que detecta um círculo como levantado é se o seu centro em relação à  $y$  está localizado a mais de 20 *pixels* de distância acima da média calculada para os eixos já detectados de cada veículo. Essa definição torna a regra dinâmica, a fim de prever que diferentes veículos terão tamanhos diferentes de eixo.

## 6.5 Representação e descrição

Com o intuito de validar se os círculos detectados a partir das etapas anteriormente descritas são referentes a um pneu ou roda de um veículo, foi obtido descritores das imagens dos círculos identificados. Subsequentemente, esses descritores foram repassados a um

classificador para treinamento e reconhecimento. Isso se faz necessário, pois a transformada de *Hough* não consegue distinguir, dentre os círculos detectados pela mesma, se se referem a um eixo de veículo ou não.

Para cumprir tal requisito, foi utilizado o descritor de imagens histograma de gradientes orientados que é disponibilizado pela biblioteca *scikit-image* (WALT et al, 2014). Os parâmetros informados para a biblioteca são a imagem que se deseja obter o descritor, o número de orientações de borda, com valor 18, número de *pixels* por célula sendo 10x10 o valor informado e o número de células por bloco tendo o valor 1x1. Também foi ativada a normalização da imagem através da utilização de raiz quadrada sobre os valores de *pixel* antes de aplicar o descritor. Os parâmetros foram definidos através de tentativa e erro buscando qual a configuração que melhor se ajusta ao classificador desenvolvido, que é descrito na sessão 6.6.

A área da imagem em que o descritor foi aplicado se baseou no raio dos círculos detectados, foi criada uma região de interesse delimitada por um retângulo que envolve o círculo, assim como demonstrado no *frame* cinza presente na figura 18. Sobre esse retângulo é aplicado o descritor de imagens digitais histograma de gradientes orientados, para se retirar as características visuais de seu conteúdo.

A imagem a seguir, demonstra uma representação visual da aplicação do descritor em imagens contendo um eixo de veículo.

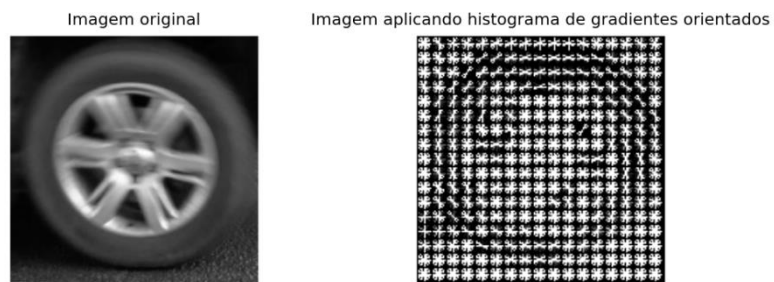
Figura 20 - Comparação entre imagem de um eixo de caminhão com o seu devido resultado após a aplicação do histograma de gradientes orientados.



Fonte: do autor.

A figura 21 também demonstra a representação retornada pela função do descritor, mas dessa vez a mesma recebe como entrada uma imagem de um eixo de um carro.

Figura 21 - Comparação entre imagem de um eixo de um carro com o seu devido resultado após a aplicação do histograma de gradientes orientados.



Fonte: do autor.

## 6.6 Reconhecimento e interpretação

Utilizando-se do descritor histograma de gradientes orientados descrito no item 6.5, um classificador foi treinado. Os seus dados de treinamento foram o resultado do descritor aplicado sobre imagens de eixos de veículos, com o intuito de reconhecer se os círculos detectados pela solução proposta realmente são eixos de veículos. O classificador escolhido se trata de uma máquina de vetor de suporte que está presente na biblioteca scikit-learn (PEDREGOSA et al, 2011), a qual foi utilizada neste projeto.

Durante a primeira versão deste trabalho havia sido definido a utilização do classificador redes neurais artificiais (ARBIB, 1995), visto que atualmente o mesmo está ganhando muito destaque devido às pesquisas realizadas sobre *deep learning*. Essa tecnologia tem um enfoque grande em problemas de classificação de objetos em imagens, pois a mesma é maleável o suficiente para apreender a classificar diversas classes de objetos e obter resultados muito bons na classificação dos mesmos. Para conseguir esses resultados a mesma é constituída de uma arquitetura complexa e necessita uma grande quantidade de dados de treinamento para que aprenda a classificar um determinado padrão de objeto. Inicialmente essa tecnologia foi vista como sendo uma boa candidata a ser utilizada neste trabalho, porém, com o avanço dos estudos sobre o tema, notou-se que a mesma tem um enfoque diferente, visa classificar vários tipos de objetos distintos recebendo uma grande quantidade de dados para treinamento de cada tipo. Este trabalho, porém, visa identificar um único objeto, eixos de veículos que trafegam em rodovias. Visto esse requisito, se visou à busca por um classificador menos complexo em arquitetura e que não necessitava uma grande quantidade de dados para o seu treinamento, sendo escolhido o classificador máquina de vetor de suporte por também

ser muito utilizado em problemas de classificação de objetos em imagens digitais, conforme as pesquisas realizadas de trabalhos relacionados, contidas no capítulo 5.

As imagens utilizadas como entradas para o descritor e conseqüentemente para o classificador foram obtidas de seis vídeos gravados no pedágio utilizado como estudo de caso. As imagens continham somente um eixo de veículo cada e foram recortadas dos vídeos de forma manual. Um total de 1.618 imagens de eixos de veículos e 669 imagens de outros objetos que não eram eixos foram utilizadas para o treinamento do classificador. As etapas definidas para o treinamento foram:

- Carregar todas as 2.287 imagens de treinamento em memória.
- Transformar as imagens para escala cinza.
- Redimensionar todas as imagens para o tamanho 32 x 32 *pixels*. Este tamanho foi definido a partir de testes realizados na aplicação, visando que o mesmo não compromettesse a eficácia do classificador.
- Obter o histograma de gradientes orientados de cada imagem.
- Repassar o resultado do gradiente ao classificador junto com a definição se o objeto é um eixo de veículo ou não.

Para que fossem encontrados os melhores parâmetros de configuração do classificador, foi utilizada a função *GridSearch* disponível na biblioteca *scikit-learn*. A função estima qual a configuração de parâmetros que leva o classificador a apreender e classificar da melhor maneira possível os dados de treinamento. A função utiliza *cross validation* para fazer a estimativa de acertos e erros do classificador, foram realizadas cinco divisões distintas dos dados de entrada para a avaliação do mesmo. Os parâmetros encontrados que melhor se adequam ao problema são um *kernel* linear e a penalidade ao classificador por erros tendo valor um. Com essa configuração, obteve-se durante o treinamento uma acurácia de 100%, tendo como desvio padrão o valor de 1%. O que demonstra que com uma divisão linear sobre os dados de entrada, o classificador conseguiu classificar corretamente a maioria dos dados utilizados no treinamento.

O momento definido para que o círculo detectado fosse verificado pelo classificador criado foi na etapa de contabilização do eixo. Onde no *frame* anterior o centro do círculo detectado deve estar anterior em relação à  $x$  à linha vertical utilizada para contagem de eixos, já no *frame* atual, o centro do círculo deve estar posterior à linha vertical. Neste momento, será aplicado o histograma de gradientes orientados para retirada das características da imagem e o resultado do histograma será repassado ao classificador máquina de vetor de



suporte. Portanto, cada eixo será analisado pelo classificador somente uma vez, caso o mesmo seja classificado como um eixo válido será feita a verificação se o mesmo está levantado ou encostado no asfalto e contabilizado o tipo de eixo detectado.

## 7 VALIDAÇÃO DO CLASSIFICADOR

Este capítulo descreve as análises realizadas com o objetivo de constatar a acurácia do classificador empregado para classificar os círculos detectados assim como medir o seu desempenho.

Os recursos computacionais utilizados para aplicar os testes foram um *notebook* com as seguintes especificações de *hardware*:

- Processador Intel® Core™ i7 Quad Core Mobile Processor I7-4710MQ (2.50GHz) 6MB
- Memória 16GB Kingston DDR3 1600MHZ (2x8GB)

As análises de desempenho do código fonte foram feitas através do uso da biblioteca cProfile presente nas bibliotecas padrão do Python. Todos os testes foram realizados no sistema operacional Ubuntu 15.10 versão 64 bits.

Foram utilizados duas versões do classificador criado para identificar se uma determinada imagem contém um eixo de um veículo. A primeira versão foi treinada com 1618 imagens de rodas de carros e caminhões e com 669 imagens de objetos de outras classes que não são eixos de veículos. A segunda versão foi treinada com 100 imagens de eixos e 40 de não eixos. As validações realizadas foram feitas em ambas as versões, com o intuito de avaliar a plasticidade do classificador sendo treinado por uma grande quantidade de dados e por um número limitado de dados de treinamento.

Para validação das duas versões do classificador criado, foram definidas cinco técnicas distintas com diferentes formas de manuseio das imagens de entrada. A seguir, irá ser detalhada cada uma dessas cinco técnicas e posteriormente apresentado os resultados obtidos com cada técnica e com cada versão do classificador.

### 7.1 Imagens que contém somente rodas

Para este teste foram utilizadas 100 imagens que contém somente rodas de veículos. Esse conjunto de dados é muito parecido com o utilizado para o treinamento do classificador, porém, essas 100 imagens não foram empregadas no treinamento do classificador. Todas as imagens foram manualmente recortadas para conterem somente o eixo de um único veículo nas mesmas. O objetivo deste teste é avaliar o classificador em um conjunto de dados parecido com o qual foi utilizado para o seu treinamento.

## 7.2 Imagens que contém uma região de interesse

Conforme o capítulo 6, a solução proposta não consegue segmentar de forma precisa a roda dos veículos que atravessam os *frames* analisados. Pois as detecções através da transformada de *Hough* detectam a roda do veículo como um círculo válido ou em certos casos, a calota do veículo. Por isso, foi definida uma região de interesse adicional ao círculo detectado. Inicialmente, foi definida como 50 *pixels* de distância das laterais e partes superior e inferior do círculo detectado, construindo assim um quadrado que envolve o círculo detectado.

Foram utilizadas 100 imagens constituídas das regiões de interesse definidas envolvendo um círculo detectado cada. O objetivo deste teste é avaliar o comportamento do classificador recebendo dados diferentes dos quais o mesmo foi treinado para classificar. Nenhuma dessas 100 imagens foi utilizada no treinamento do classificador.

## 7.3 Imagens que contém uma região de interesse com aplicação de *sliding window*

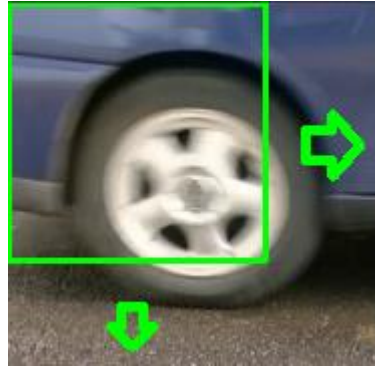
Da mesma maneira do item 7.2, este teste também utiliza o mesmo conjunto de 100 imagens contendo a região de interesse que envolve o círculo detectado, porém, utilizando a técnica de *sliding window*. Essa técnica é utilizada em problemas de classificação de objetos em imagens digitais através da definição de um quadrado ou retângulo de tamanho fixo que desliza sobre os *pixels* presentes na imagem. Foi definido um quadrado de 128 de largura por 128 *pixels* de comprimento.

Para que não sejam percorridos e analisados todos os *pixels* da imagem, é definido uma quantidade de *pixels* que se deseja pular a cada iteração da técnica, este número foi definido em 8 *pixels*. A cada quadro analisado do tamanho pré-fixado, o retângulo irá pular 8 *pixels* para a direita, contendo assim uma nova representação da imagem original. Isto é feito até que não seja mais possível deslizar a janela para a direita, então, a janela é deslizada de volta a posição 0 em relação a x e desliza 8 *pixels* para baixo em relação a y. Desta maneira, é possível diminuir bastante o número de análises feitas em comparação com uma análise *pixel* a *pixel*, onde o deslize tem tamanho de 1 *pixel*.

O objetivo dessa análise é não repassar a imagem da região de interesse de forma integral ao classificador, mas de forma parcial, avaliando assim, qual a porcentagem de janelas analisadas resultou na classificação como roda e não roda pelo classificador. A figura

22 demonstra a aplicação da técnica *sliding window*, as duas setas verdes inseridas na imagem servem para exemplificar o movimento realizado pelo deslize da região.

Figura 22 - Exemplo de aplicação da técnica *sliding window* sobre uma região de interesse.

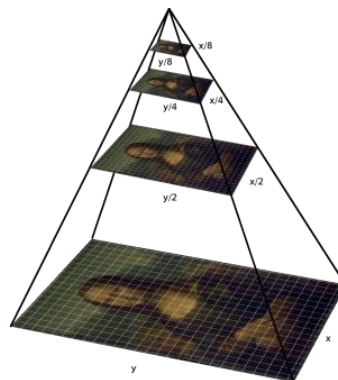


Fonte: do autor.

#### 7.4 Imagens que contém uma região de interesse com aplicação de *sliding window* e *image pyramid*

Esta análise utiliza-se do mesmo conjunto de dados que foi utilizado nos itens 7.2 e 7.3, além de também utilizar a técnica de *sliding window* aplicada no item anterior. A principal adição dessa validação é a utilização da mudança da escala da imagem da região de interesse após a utilização sucessiva da técnica de *sliding window*. A imagem 23 exemplifica a mudança de escala realizada em uma imagem, diminuindo o seu tamanho original.

Figura 23 - Exemplo de variação de escala em imagens através de *image pyramid*.



Fonte: IIPImage. Disponível em <<http://iipimage.sourceforge.net/documentation/images/>>. Acesso em:

19/09/2016.

Nos casos onde a transformada de *hough* detecta a calota do veículo como um círculo válido e a calota se refere a uma roda de caminhão, a definição da região de interesse acaba contemplando a roda inteira do caminhão. Portanto a delimitação de uma área fixa imposta pela técnica *sliding window* não se torna atrativa, pois a mesma nunca consegue conter a roda do caminhão inteira. A imagem a seguir demonstrada um exemplo deste caso, onde a região de interesse cobre a roda inteira de um caminhão.

Figura 24 - Exemplo de imagem de região de interesse que contempla toda a roda de um caminhão.



Fonte: do autor.

Por este motivo, a técnica *image pyramid* altera a escala da imagem original após cada iteração completa da técnica *sliding window* sobre a imagem. Este processo de alteração de escala é finalizado quando a imagem original tem um tamanho menor do que a do quadrado definido na técnica *sliding window*.

Com essa validação, pretende-se avaliar a porcentagem de janelas que foram classificadas como roda, levando em consideração as mudanças de escala realizada sobre cada imagem analisada.

### **7.5 Imagens que contém uma região de interesse de tamanho variável**

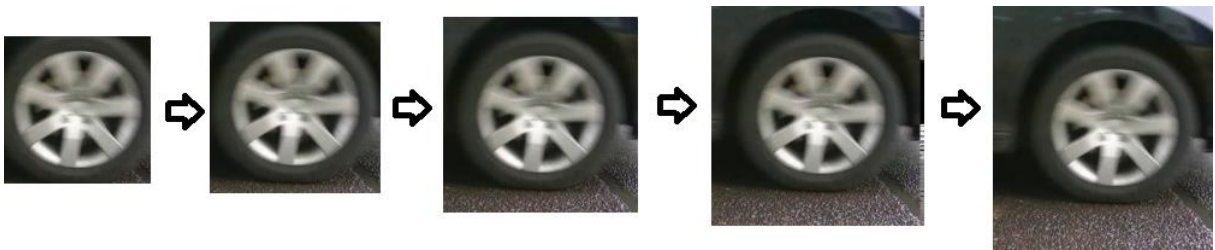
Uma alternativa quanto à incerteza de uma determinada região de interesse enquadrar ou não de forma correta a totalidade de uma roda de um veículo é através da variação do tamanho da região de interesse. No item 7.2 a região de interesse foi delimitada de forma fixa,

50 *pixels* de distância da borda do círculo detectado pela transformada de *Hough*. Desta vez, o objetivo é obter diferentes regiões de interesse sobre cada imagem analisada.

Nesta análise, pretende-se fazer diferentes delimitações, definidas em 10, 20, 30, 40 e 50 *pixels* de distância do círculo detectado. Cada uma dessas diferentes delimitações é repassada ao classificador. Caso seja identificada como uma roda válida em quaisquer das delimitações, a imagem é definida como sendo uma roda. Foram utilizadas 119 imagens nesta etapa, sendo que nenhuma foi utilizada para o treinamento do classificador.

A imagem 25 demonstra os diferentes tamanhos de região de interesse aplicados sobre um mesmo eixo, começando em 10 *pixels* de distância da borda do círculo detectado pela transformada de *Hough* e chegando até em 50 *pixels* de distância.

Figura 25 - Exemplo de variação da região de interesse sobre um círculo detectado pela transformada de *Hough*.



Fonte: do autor.

## 7.6 Avaliação do classificador

A tabela 2 contém os resultados obtidos pelas duas versões do classificador, treinado com muitos dados ou com poucos, sobre os cinco tipos diferentes de técnicas para manusear os dados utilizados como entrada para os testes de classificação.

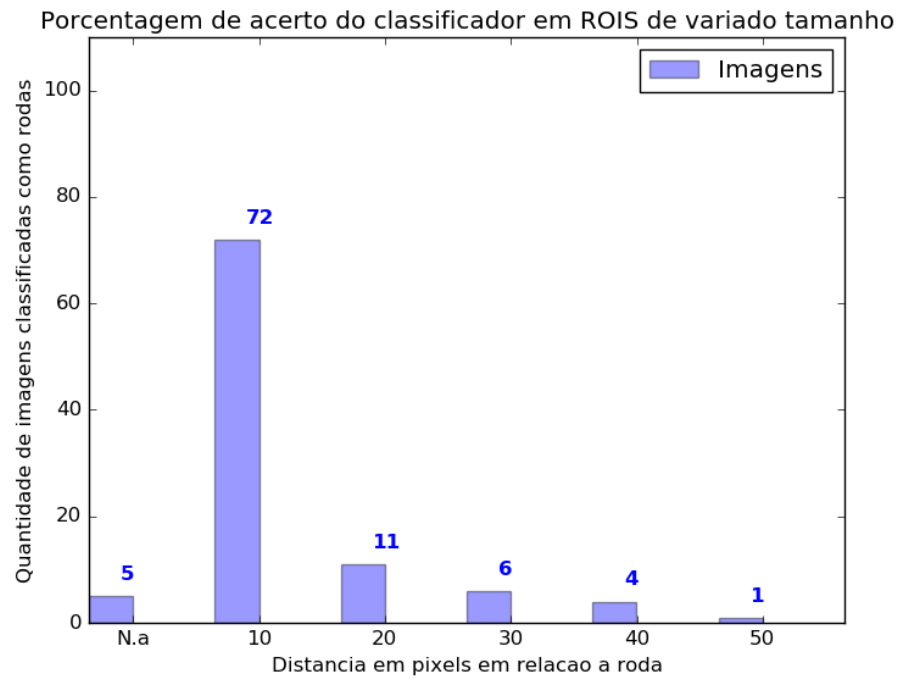
Tabela 2 - Resultados obtidos pelo classificador utilizado.

Tipo de treinamento do classificador	Análise realizada	Quantidade de vezes que o classificador foi executado	Tempo de processamento	Porcentagem de acerto
Treinado com muitos dados	Imagens somente rodas	100	0,045 segundos	87%
Treinado com poucos dados	Imagens somente rodas	100	0,043 segundos	89%
Treinado com muitos dados	Imagens ROI	100	0,046 segundos	86%
Treinado com poucos dados	Imagens ROI	100	0,044 segundos	92%
Treinado com muitos dados	Imagens ROI com <i>sliding window</i>	20.044	7,797 segundos	21%
Treinado com poucos dados	Imagens ROI com <i>sliding window</i>	20.044	7,884 segundos	33%
Treinado com muitos dados	Imagens ROI com <i>sliding window</i> e <i>image pyramid</i>	33.531	13,781 segundos	44.99%
Treinado com poucos dados	Imagens ROI com <i>sliding window</i> e <i>image pyramid</i>	33.531	13,231	53.24%
Treinado com muitos dados	Imagens com tamanho de ROI variável	163	0.170 segundos	95%
Treinado com poucos dados	Imagens com tamanho de ROI variável	147	0,163 segundos	94%

Fonte: do Autor.

Conforme os resultados obtidos na tabela 3, obtiveram-se os melhores percentuais de acerto através da utilização do classificador que foi treinado por um conjunto maior de dados, além disso, utilizando sobre os dados de validação a variação da região de interesse que encobre o eixo de veículo. Essa foi a configuração adotada para classificar os círculos detectados pela transformada de *Hough* neste trabalho. O gráfico presente na figura 26 demonstra a quantidade de imagens classificadas como rodas em cada variação da região de interesse utilizada, os dados referem-se à utilização do classificador treinado com mais dados.

Figura 26 – Gráfico da quantidade de imagens classificadas como roda nas diferentes variações da região de interesse.



Fonte: do autor.

Observa-se que a grande maioria das imagens foram classificadas corretamente na região de interesse de tamanho 10, essas imagens possivelmente referem-se a eixos de carros, pois os mesmos não necessitam de uma região de interesse muito grande para que a mesma consiga encobrir o eixos do veículo.



## 8 RESULTADOS DO SISTEMA PROPOSTO

Para realizar a análise da solução proposta, uma nova coleta de vídeos foi realizada no pedágio utilizado como estudo de caso, a mesma foi realizada no dia 05/10/2016. Foram coletados 14 vídeos contendo 24 caminhões e um ônibus, além de 38 carros. Todos os 14 vídeos capturados foram repassados ao sistema proposto para que fosse contabilizada a quantidade de eixos de cada veículo que estão encostados no asfalto da rodovia e quantos estão suspensos.

### 8.1 Desempenho

Visando avaliar o desempenho das análises realizadas pelo sistema proposto e utilizando-se dos recursos de *hardware* descritos no item 7, foi realizada a medição do tempo de processamento sobre cada função presente no código fonte da solução proposta, assim como uma medição geral da execução completa. O vídeo utilizado neste teste continha 5 minutos e 10 segundos de duração, sendo que nele houve a passagem de 10 carros e 3 caminhões.

O tempo utilizado para analisar os 9.281 *frames* do vídeo foi de 42,953 segundos, resultando em uma média de aproximadamente 216 *frames* analisados por segundo. As funções que mais demandaram tempo de processamento foram:

- **Redimensionar os *frames* analisados:** 18,007 segundos
- **Aplicar *Gaussian Blur*:** 9,021 segundos
- **Capturar os *frames* do vídeo para memória:** 7,277 segundos
- **Conversão para escala cinza:** 1,647 segundos
- **Procura por contornos:** 1,640 segundos
- **Aplicação da transformada de *Hough*:** 1,118 segundos
- **Acúmulo da diferença entre o *frame* atual e o anterior:** 1,022 segundos

Uma segunda análise foi realizada para demarcar o tempo de execução do programa em relação à entrada e saída de um único veículo, onde o primeiro *frame* analisado corresponde à entrada do veículo no vídeo e o último a saída do mesmo. Este teste teve análise de 96 *frames*, sendo que o tempo de execução total foi de 0,565 segundos, resultando

em uma média de 169 *frames* analisados por segundo. Nessa análise, as funções que mais demandaram tempo de processamento foram:

- **Redimensionar os *frames* analisados:** 0, 213 segundos
- **Aplicar *Gaussian Blur*:** 0,096 segundos
- **Aplicação da transformada de *Hough*:** 0,090 segundos
- **Capturar os *frames* do vídeo para memória:** 0,076 segundos
- **Procura por contornos:** 0.022 segundos
- **Conversão para escala cinza:** 0.018 segundos

## 8.2 Exatidão nas análises de eixo

Para a contabilização das estatísticas de acertos e erros do sistema proposto sobre os eixos encostados no asfalto e suspenso de cada veículo, foram analisados todos os 14 vídeos coletados para validação. A seguir, é apresentada a tabela dos resultados obtidos.

Tabela 3 - Resultados obtidos pelo sistema proposto.

	Quantidade
Eixos de caminhão/ônibus detectados corretamente	54
Eixos de caminhão/ônibus detectados incorretamente	16
Média de acerto	77%
Eixos levantados de caminhão/ônibus detectados corretamente	6
Eixos levantados de caminhão/ônibus detectados incorretamente	3
Média de acerto	66%
Eixos de carros detectados corretamente	75
Eixos de carros detectados incorretamente	1
Média de acerto	98%
Porcentagem de caminhões/ônibus com todos os eixos detectados corretamente	52%
Porcentagem de carros com todos os eixos detectados corretamente	97%

Fonte: do Autor.

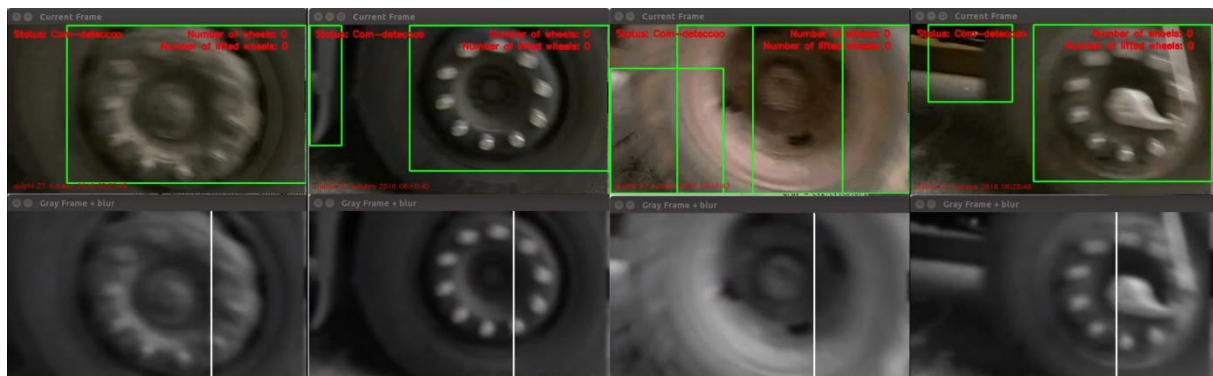
Dentre os eixos detectados incorretamente, as causas para cada falha na detecção será analisada a seguir, incluindo imagens do momento em que o erro aconteceu. As imagens utilizadas para exemplificar o ocorrido foram divididas em duas regiões, no conjunto superior destaca-se a detecções de contornos pela solução proposta, demarcada pelos retângulos verdes. Já no inferior, encontram-se os círculos detectados pela transformada de *Hough*, a linha vertical utilizada para contagem de eixos, a linha horizontal utilizada para validação dos

eixos levantados e também a demarcação máxima da região de interesse que envolve os círculos detectados, assim como descrito no item 7.5.

### 8.2.1 Eixos não detectados pela transformada de *Hough*

Quatro eixos de caminhões que estavam encostados no asfalto não foram detectados pela transformada de *Hough*, apesar de a detecção de contornos ter englobado o eixo do veículo. Abaixo, é apresentada a imagem contendo os eixos que falharam nesta detecção.

Figura 27 - Quatro eixos que falharam na detecção de círculos pela transformada de *Hough*.



Fonte: do autor.

Observa-se, que existem círculos dentro da área demarcada pelos contornos de todos os quatro eixos que estão nesta categoria. Esses círculos referem-se à calota dos eixos dos caminhões, porém estes não foram considerados círculos aptos para detecção pela transformada de *Hough*. Possivelmente, pois as bordas dos círculos não foram bem identificadas pela transformada devido às calotas dos eixos um dois e quatro, presentes na imagem 27, estarem pintados da cor preta, dificultando a definição clara dos contornos do objeto.

### 8.2.2 Eixo sem os contornos bem definidos

Um eixo encostado no asfalto não teve os contornos bem detectados pela solução proposta. Na figura 28, é apresentada a imagem do eixo que se encontra nessa categoria.

Figura 28 - Eixo onde os contornos detectados foram muito pequenos, não englobando todo o eixo do veículo.



Fonte: do autor.

A calota deste eixo estava pintada de bordô, em um tom parecido com o preto do pneu do veículo, ocasionando com que não ocorresse tanta variação de cor conforme este eixo se deslocava pelos *frames* do vídeo analisado. A partir da pouca variação de cor e de luz, a diferença entre o *frame* atual e o anterior é pequena, fazendo com que somente pequenas áreas sejam detectadas no *frame* delta e consequentemente nos contornos detectados.

### 8.2.3 Eixos levantados não classificados como rodas

Três eixos levantados não foram classificados como eixos de veículos válidos pelo classificador utilizado. A imagem a seguir, demonstra o momento em que cada um dos eixos foi repassado ao classificador para validar a sua contabilização para a contagem de eixos do veículo.

Figura 29 - Três círculos que não foram validados pelo classificador como sendo eixos válidos de um veículo.



Fonte: do autor.

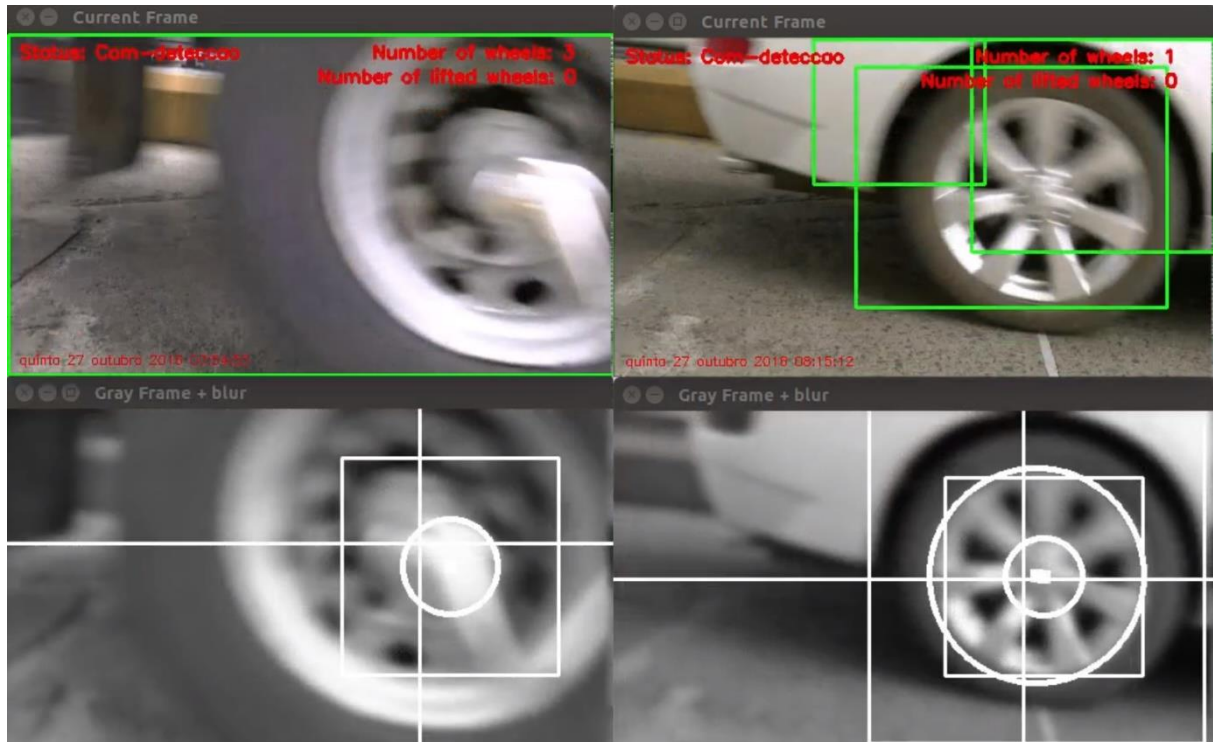
Em todos os três casos, o círculo detectado pela transformada de *Hough* é pequeno e está centralizado no centro da calota dos eixos. A área delimitada pelo retângulo branco que envolve os círculos é pequena e não engloba todo o eixo do veículo, resultando em uma área da imagem que não compreende um eixo válido a ser repassado para o histograma de gradientes orientados e posteriormente ao classificador.

Destaca-se também que a filmagem realizada não capturou por completo o eixo de caminhões, que tem um tamanho maior por natureza. Isso porque a filmadora utilizada não tinha uma abertura de captura grande e estava posicionada relativamente próxima a rodovia. Esse fato contribuiu para que os dados analisados para a validação da solução proposta fossem incompletos, dificultando as análises realizadas.

#### 8.2.4 Eixos encostados no asfalto não classificados como rodas

Dois eixos que estavam no asfalto não foram classificados como eixos válidos de veículos pelo classificador. A imagem 30 contém os dois eixos que se enquadraram nesta categoria.

Figura 30 - Dois círculos de eixos encostados no asfalto que não foram validados pelo classificador como sendo eixos válidos de um veículo.



Fonte: do autor.

O caso da esquerda, onde o eixo de um caminhão não é classificado como sendo um eixo válido se enquadra na mesma situação descrita pelo item 8.2.3, onde o retângulo que envolve o círculo não contém informação suficiente para caracterizar o eixo de um veículo válido.

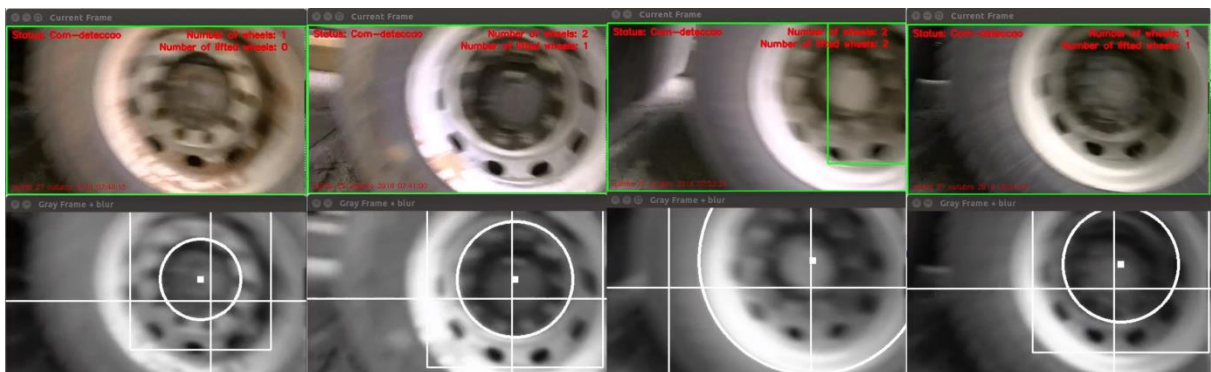
Já a situação da direita, mostra o único eixo de carro que foi contabilizado incorretamente pelo sistema proposto. Nesta situação, dois círculos foram detectados dentro da calota do mesmo eixo. O círculo maior envolve toda a calota do eixo, já o menor, encobre somente o seu centro. Apesar dos dois círculos terem sido detectados, somente um foi repassado ao classificador para contabilização de eixos para este veículo. Isso, pois a regra definida exige que para ser analisado, o círculo deve estar no *frame* anterior com o seu centro em relação à  $x$  antes da linha vertical definida e no *frame* atual posterior a linha vertical. O círculo menor se enquadrava nesta regra, já o círculo maior, que encobre toda a calota do veículo não existia no *frame* anterior, ocasionando que o mesmo não fosse avaliado pelo classificador.

Em todos os dois casos, o círculo detectado pela transformada de *Hough* é pequeno e está centralizado no centro da calota dos eixos. A área delimitada pelo retângulo branco que envolve os círculos é pequena e não engloba todo o eixo do veículo, resultando em uma área da imagem que não compreende um eixo válido a ser repassado para o classificador.

### 8.2.5 Eixos encostados no asfalto detectados como levantados

Seis eixos de caminhões que estavam no asfalto foram detectados como eixos levantados pelo sistema proposto, devido ao seu centroide em relação à  $y$  estar consideravelmente acima da média calculada para os eixos identificados anteriormente. Abaixo, é apresentada a imagem contendo quatro dos casos que se enquadraram nesta situação.

Figura 31 - Quatro eixos que foram incorretamente avaliados como estando levantados pelo sistema proposto.



Fonte: do autor.

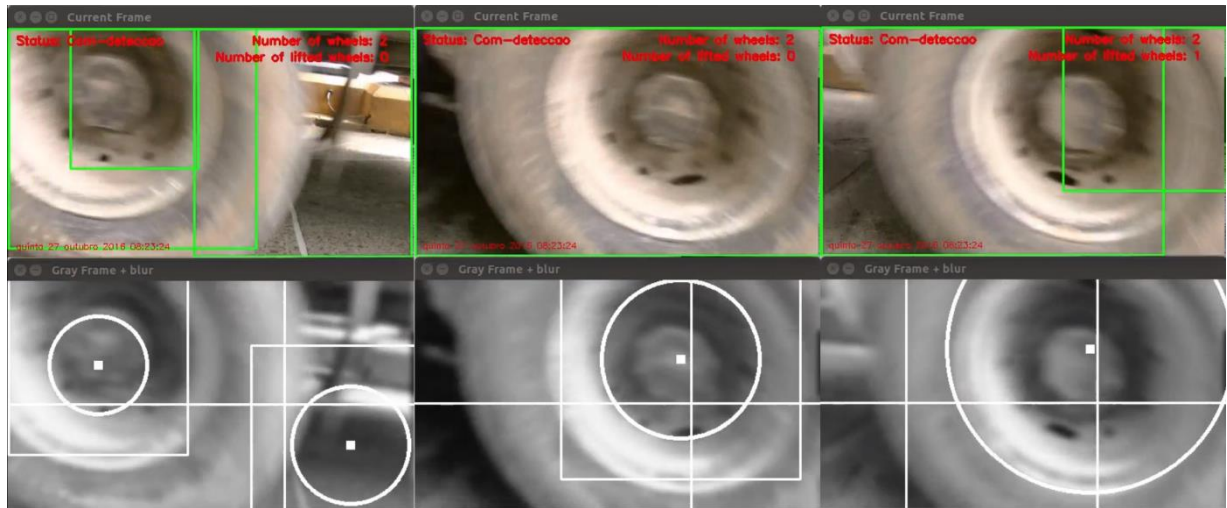
Nos casos acima o limiar de  $20 \text{ pixels}$  estabelecido como distância máxima entre o centroide de um eixo e a média calculada não foi suficiente para que os eixos fossem detectados como estando encostados no asfalto. Todos os eixos presentes na imagem acima ultrapassaram este limiar, estando superior em relação à  $y$  a mais de  $20 \text{ pixels}$  da linha horizontal.

A situação presente na imagem 32 é outra, nela, a imagem mais a esquerda exemplifica o momento em que foi contabilizado um eixo falso positivo para o veículo presente nas imagens. A partir deste eixo falso positivo detectado, a média do centroide dos eixos daquele veículo diminuiu, ficando abaixo do valor real verdadeiro dos eixos daquele veículo. As imagens do centro e da direita foram os próximos dois eixos do veículo a serem



analisados. Como consequência da interferência do eixo falso positivo detectado, os dois eixos foram detectado como estando levantados.

Figura 32 - Imagem da esquerda demonstra o eixo falso positivo contabilizado que interferiu na posição da linha horizontal. As imagens do centro e da direita são os próximos eixos do veículo avaliados como levantados erroneamente.

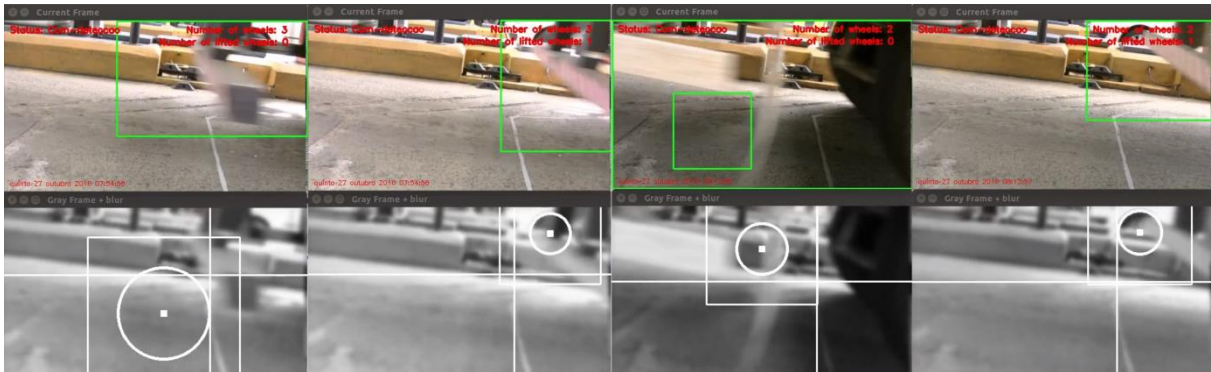


Fonte: do autor.

### 8.2.6 Eixos detectados que são falsos positivos

Quatro eixos falsos positivos foram contabilizados pelo sistema proposto. A definição de uma detecção falsa positiva adotada refere-se a um eixo que foi detectado e contabilizado como estando no asfalto ou levantado, porém na verdade não é um objeto que se refere a um eixo de veículo válido. Na imagem 33, é demonstrado dois desses casos, o primeiro e o terceiro conjunto de imagens demonstram os círculos detectados antes da linha vertical utilizada para contagem de eixos e o segundo e quarto conjunto mostram os círculos detectados posteriores à linha.

Figura 33 - Imagens demonstrando círculos que foram contabilizados como eixos sendo falsos positivos.

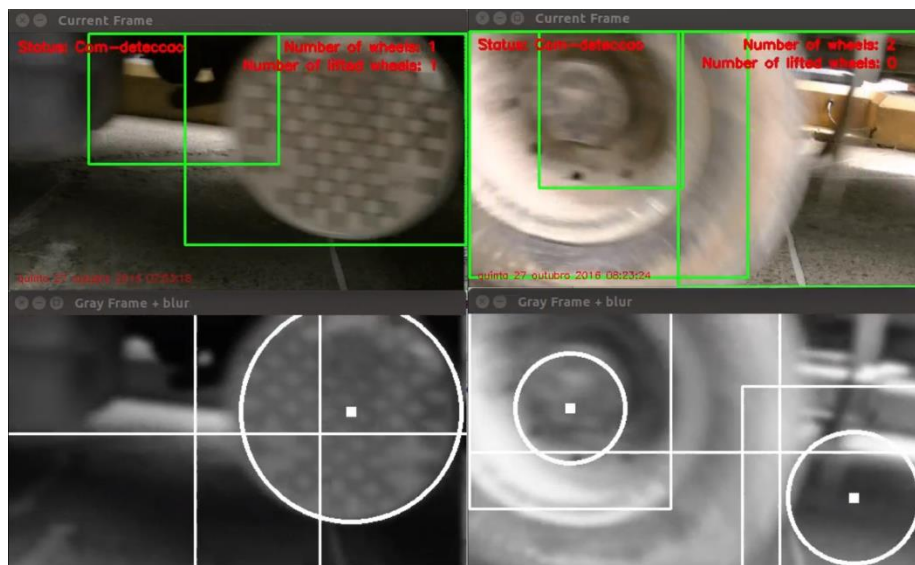


Fonte: do autor.

No segundo e quarto conjunto de imagens, os círculos foram repassados ao classificador utilizado para verificação se os mesmos são ou não eixos válidos, pois passaram na validação que verifica se no *frame* anterior os mesmos círculos estavam antes da linha vertical utilizada. Nestes dois casos, o classificador avaliou os círculos como sendo válidos incorretamente.

Na imagem a seguir, são apresentados outros dois casos de eixos falsos positivos identificados pela solução proposta. O primeiro se trata de um objeto circular presente na carroceria do caminhão, objeto este que foi avaliado como sendo um eixo válido pelo classificador. No segundo conjunto de imagens, dois círculos foram detectados no mesmo *frame* do vídeo. O círculo da direita, que não se trata de um eixo de um veículo, foi repassado ao classificador para validação, que o avaliou como sendo um círculo válido.

Figura 34 - Imagens demonstrando círculos que foram contabilizados como eixos sendo falsos positivos.



Fonte: do autor.

### 8.3 Matriz de confusão para duas classes

A partir da análise dos 14 vídeos utilizados anteriormente para validação do sistema proposto, foi montada a matriz de confusão para as duas classes utilizadas no classificador treinado, sendo elas eixos de veículos e outros objetos. A tabela a seguir demonstra a convenção utilizada para a obtenção dos resultados da matriz de confusão.

Tabela 4 - Tabela contendo a estrutura utilizada na matriz de confusão para duas classes.

Classe	Objeto foi predito como eixo de veículo	Objeto foi predito como não sendo eixo de veículo	Taxa de erro da classe	Taxa de erro total
Classe de eixos de veículo	TP	FN	$\frac{FN}{TP + FN}$	$\frac{FP + FN}{n}$
Classe de não eixos de veículo	FP	TN	$\frac{FP}{FP + TN}$	

Fonte: do Autor.

Sendo que TP representa os valores verdadeiros positivos, FN os falsos negativos, FP os falsos positivos, TN os verdadeiros negativos e n representa a soma dos valores de TP, FN, FP e TN. As taxas de erros de classe e erro total visam clarificar os casos onde objetos de uma determinada classe foram classificados como sendo de outra. A seguir, encontra-se a tabela com os resultados obtidos para a matriz de confusão para duas classes.

Tabela 5 – Resultados da matriz de confusão para duas classes.

Classe	Objeto foi predito como eixo de veículo	Objeto foi predito como não sendo eixo de veículo	Taxa de erro da classe	Taxa de erro total
Classe de eixos de veículo	116	8	0,064	0.091
Classe de não eixos de veículo	4	3	0,571	

Fonte: do Autor.

A tabela 5 demonstra que a taxa de erro total do classificador sobre os 14 vídeos utilizados para validação é de 9%, já a taxa de erro para objetos que são eixos de veículos é de 6% e para objetos que não são eixos é de 57%, resultado este que provém dos objetos falsos positivos que foram contabilizados como eixos válidos de veículos.

## 9 CONCLUSÃO

A contabilização da quantidade de eixos que cada veículo possuiu teve resultados diferentes para carros e caminhões. No primeiro grupo, a porcentagem de acerto na contabilização de eixos foi alta, estando próxima de 100%. Já o segundo teve uma média menor, chegando a 77% de acerto. Assim como foi demonstrado nas imagens analisadas anteriormente, os eixos de caminhões e ônibus não foram capturados por inteiro nas filmagens realizadas para validação da solução proposta. Para que os resultados fossem melhores, seria necessário que todo o eixo estivesse contido dentro dos *frames* e que a filmagem estivesse com o seu ângulo alinhado com a rodovia. Visando esse objetivo, seria interessante posicionar uma micro câmera na calçada da rodovia, com o seu ângulo alinhado a mesma, contendo uma lente objetiva capaz de distanciar os objetos filmados da câmera, através da utilização de um ângulo grande de captura.

A avaliação dos eixos levantados teve a mesma interferência, as análises também foram comprometidas por a filmagem realizada não ter englobado por completo os eixos dos veículos. Mesmo assim, a solução proposta obteve 66% de acerto nas análises desse tipo de eixo.

Os resultados obtidos se baseiam em capturas de dados reais, que são influenciados pela iluminação do local de captura. No caso de ambientes noturnos, onde não se fez captura de dados, os resultados tendem a não se modificarem, através da utilização do algoritmo desenvolvido, visto que o local onde foram feitas as capturas recebe boa iluminação noturna. Este ponto porém, pode também ser explorado em trabalhos futuros, visando obter filmagens em uma conjuntura maior de condições climáticas, como em ambientes noturnos, com chuva, neblina ou baixa iluminação.

Visando aplicar esta solução em um sistema embarcado, sugere-se a utilização da biblioteca SimpleCV, a mesma é escrita em Python e é voltada para atuação em *hardware* embarcado.

A técnica transformada de *Hough* trouxe resultados interessantes na detecção de círculos, porém, na maioria dos casos acabou detectando a calota do eixo do veículo. Isso porque a calota geralmente tem suas bordas mais bem definidas do que o pneu do veículo,

ocasionando que a mesma tenha uma melhor avaliação pela técnica e seja avaliada como um círculo mais interessante a ser detectado. Por esse motivo a avaliação se um determinado círculo é realmente um eixo de veículo se tornou mais complexa, visto que é difícil segmentar a área total de um determinado eixo somente pela sua calota. A utilização da região de interesse variável que engloba os círculos detectados não conseguiu segmentar a totalidade de eixos de caminhão, por o seu tamanho ser muito grande, ocupando a totalidade do *frame* das filmagens em muitos casos.

Círculos que foram contabilizados como eixos válidos de veículos, mas na verdade são falsos positivos, demonstram que a técnica histograma de gradientes orientados não foi suficiente para descrever com precisão um eixo de veículo válido. Para uma melhor representação, acredita-se que devem ser considerados uma conjuntura maior de fatores, como textura, cor e formato, não somente a orientação das bordas presentes na imagem, como é o caso da técnica utilizada. Esse ponto pode ser abordado em trabalhos futuros, com a utilização de outros descritores de imagem que utilizem mais fatores para a descrição realizada.

Sugere-se também, como trabalhos futuros, a validação de uma conjuntura maior de diferentes descritores de imagens e classificador de objetos em imagens digitais, visto que certos classificadores obtém melhores resultados dependendo da estrutura dos dados utilizados em seu treinamento e validação. Também podem ser exploradas diferentes técnicas de mineração de dados, como busca de *outliers* nos dados de treinamento e redução da dimensionalidade.

## REFERÊNCIAS

- ARAGÓN, A. C. Vision-based detection and labelling of multiple vehicle parts. 2011. Disponível em: <<http://www.site.uottawa.ca/~laganier/publications/chavez.pdf>>. Acesso em: 20 abr. 2016.
- ARBIB, MICHAEL A. *The Handbook of Brain Theory and Neural Networks*. 2. ed. Reino Unido: The MIT Press, 1995.
- BANDARU, K; PATIEJUNAS, K. 2015. *Under the hood: Facebook's cold storage system*. Disponível em: <<https://code.facebook.com/posts/1433093613662262/-under-the-hood-facebook-s-cold-storage-system-/>>. Acesso em: 21 abr. 2016.
- DALAL, N.; TRIGGS, B. *Histograms of Oriented Gradients for Human Detection*. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). Estados Unidos: IEEE, 2005.
- DEAN, J; MONGA, R. 2015. *TensorFlow* - Google's latest machine learning system, open sourced for everyone. Disponível em: <[http://googleresearch.blogspot.com.br/2015/11/tensorflow-googles-latest-machine\\_9.html](http://googleresearch.blogspot.com.br/2015/11/tensorflow-googles-latest-machine_9.html)> Acesso em: 21 abr. 2016.
- EVANS, B. *How many pictures?*. Disponível em: <<http://ben-evans.com/benedictevans/2015/8/19/how-many-pictures>>. Acesso em: 21 abr. 2016.
- FRIEDMAN, J. H. *Greedy function approximation: a gradient boosting machine*. Annals of statistics, JSTOR, p. 1189–1232, 2001.
- GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. 3.ed. Estados Unidos: Prentice Wiley-Interscience, 2007.
- GUO G.; WANG H.; BELL D.; BI Y.; GREER K. Knn model-based approach in classification. *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*. Local:Springer, 2003.
- GOUTTE C.; GAUSSIER E.; A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. *Advances in Information Retrieval*. Espanha:Springer, 2005.
- HAN, J. *Data Mining: Concepts and Techniques*. 3.ed. Estados Unidos: Morgan Kaufmann Publishers Inc, 2011.
- HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The elements of statistical learning: data mining, inference and prediction*. 2. ed. Estados Unidos: Springer, 2009.
- HEISELE, B; HO, P; POGGIO, P. *Face Recognition with Support Vector Machines: Global versus Component-based Approach*. Massachusetts Institute of Technology Center for Biological and Computational Learning, 2001. Disponível em <<http://cbcl.mit.edu/cbcl/publications/ps/iccv2001.pdf>>. Acesso em 28 de setembro de 2016.
- HOSMER, D. W.; LEMESHOW, S. *Applied Logistic Regression*. 2. ed. Estados Unidos: Wiley, 2000.

- MARSLAND, S. *Machine Learning: An algorithm perspective*. 2.ed. Chapman & Hall/CRC, 2015.
- MASON, L.; BAXTER, J; BARTLETT, P.; FREAN, M. *Boosting algorithms as gradient descent*. Estados Unidos:MIT Press, 2000.
- MESSELODI, S. MODENA, C. M., CATTONI. G. Vision-based bicycle/motorcycle classification. 2007. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0167865507001377>>. Acesso em: 20 abr. 2016.
- NG. A.; JORDAN, M. *On Discriminative vs. Generative classifiers: A comparison of Logistic Regression and Naïve Bayes*. Neural Information Processing Systems, 2001.
- OTSU, N. (1979). *A threshold selection method from gray-level histograms*, *IEEE Transactions on Systems, Man and Cybernetics* 9(1): 62–66, 1979.
- Open Source Computer Vision Library (OPENCV)*. Disponível em: <<http://opencv.org/about.html>>. Acesso em: 07 maio 2016.
- PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCJESNAY, E; *Scikit-learn: Machine Learning in {P}ython*. Journal of Machine Learning Research, páginas 2825-2830. (2011).
- PERMUTER, H. ; FRANCOS, J. ; JERMYN I. H. *Gaussian mixture models of texture and colour for image database retrieval*. IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003.
- RABINER, L. R. *A tutorial on Hidden Markov Models and selected applications in speech recognition*. *Proceedings of the IEEE*. Estados Unidos: IEEE, 1989.
- SANTOS, T T. Detecção automática de bagas de café em imagens de campo. Disponível em: <<http://ainfo.cnptia.embrapa.br/digital/bitstream/item/131791/1/22-thiago-teixeira-santos-162.pdf>>. Acesso em: 20 abr. 2016.
- SAID, Y., ATRI, M., TOURKI, R. *Human detection based on integral histograms of oriented gradients and svm*. In *Communications, Computing and Control Applications (CCCA)*, 2011 International Conference, páginas 1 –5. (2011).
- SCHMID,C.; MOHR, R *Combining gray-value invariants with local constraints for object recognition*. *Conference on Computer Vision and Pattern Recognition*, pp. 872–877, 1996.
- SILVA, D.; QUINTANA L.; MONGELO A.; CEREDA M.; PISTORI H. Classificação de leveduras utilizando transformada de Hough e aprendizagem supervisionada. WVC 2012 - Workshop de Visão Computacional, 27-30 Maio, Goiânia, Goiás, 2012.
- SILVA, L. A. Classificação de nódulos mamográficos utilizando um comitê de redes neurais artificiais. São Paulo: Escola Politécnica, Universidade de São Paulo, 2005. Dissertação de Mestrado em Sistemas Eletrônicos. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/3/3142/tde-05092006-203231/>>. Acesso em: 15 nov. 2015.

- SILVA, R.; AIRES K.; SANTOS T.; ABDALLA K.; VERAS R. Segmentação, classificação e detecção de motociclistas sem capacete. 2013. Disponível em: <<http://www.sbai2013.ufc.br/pdfs/5169.pdf>>. Acesso em: 20 abr. 2016.
- SIVARAMAN, S. *Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior*. Disponível em: <[http://cvrr.ucsd.edu/publications/2013/SayananTrivedi\\_IEEETITS2013\\_VehicleSurvey.pdf](http://cvrr.ucsd.edu/publications/2013/SayananTrivedi_IEEETITS2013_VehicleSurvey.pdf)>. Acesso em: 20 abr. 2016.
- SIVIC, J; ZISSERMAN, A. *Video Google: A text retrieval approach to object matching in videos*. IEEE International Conference on Computer Vision, Nice, France, 2003, pp. 1470–1477.
- SOBEL, I. E. *Camara Models and Machine Perception, Ph.d dissertation*, Stanford University, Palo Alto, Calif. 1970
- TOLA, E.; LEPETIT, V.; FUA, P. *Daisy: An efficient dense descriptor applied to wide-baseline stereo*. IEEE Trans. Pattern Anal. Mach. Intell. Estados Unidos: IEEE Computer Society, 2010.
- TUYTELAARS, T.; MIKOLAJCZYK, K. *Local Invariant Feature Detectors: A Survey*. Estados Unidos: Now Publishers Inc, 2007.
- UTGOFF, P. E. *Incremental induction of decision trees. Machine Learning*. Estados Unidos: Springer Verlag, 1989.
- VAPNIK V. *Estimation of Dependences Based on Empirical Data*. Nauka, Moscow. [in Russian] (English translation: 1982, Springer Verlag, New York), 1979.
- VAPNIK V. *The nature of Statistical Learning Theory*. Estados Unidos: Springer Verlag, 1995.
- WALT, V. D. S.; SCHÖNBERGER, L. J.; NUNEZ-IGLESIAS, J.; BOULOGNE, F; WARNER, D. J.; YAGER, N.; GOUILLART, E.; YU, T. scikit-image: Image processing in Python, 2014. Disponível em: <<https://peerj.com/articles/453/>>. Acesso em: 21 setembro 2016.
- YUEN H. K.; ILLINGWORTH J.; KITTLER J. *Detecting partially occluded ellipses using the Hough transform*. *Image and Vision Computing*, vol. 7, no. 1, pp. 31-37, 1989.
- ZIVKOVIC, Z. *Improved adaptive gaussian mixture model for background subtraction*. International Conference on Pattern Recognition 2004, Vol. 2, pp. 28 – 31. 2004.
- ZHOU, Y; CHEUNG, N. *Vehicle classification using Transferable Deep Neural Networks Features*. Disponível em: <<http://arxiv.org/abs/1601.01145>>. Acesso em: 03 maio 2016.