

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MARCELL MANFRIN BARBACENA

**IMPACTO DA REDUÇÃO DA TAXA DE TRANSMISSÃO DE FLUXOS
DE VÍDEOS NA EFICÁCIA DE ALGORITMO PARA DETECÇÃO DE
PESSOAS**

Campina Grande
2014

MARCELL MANFRIN BARBACENA

**IMPACTO DA REDUÇÃO DA TAXA DE TRANSMISSÃO DE FLUXOS
DE VÍDEOS NA EFICÁCIA DE ALGORITMO PARA DETECÇÃO DE
PESSOAS**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande como requisito para a obtenção do título de Mestre.

Orientador: Prof. Dr. Francisco Vilar Brasileiro

Co-orientador: Prof. Dr. Herman Martins Gomes

Área de concentração: Visão computacional

Campina Grande
2014

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

B228i Barbacena, Marcell Manfrin.
Impacto da redução da taxa de transmissão de fluxos de vídeos na eficácia de algoritmos para detecção de pessoas / Marcell Manfrin Barbacena. – Campina Grande, 2014.
82 f. : il. color.

Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2014.

"Orientação: Prof. Dr. Francisco Vilar Brasileiro, Prof. Dr. Herman Martins Gomes".
Referências.

1. Visão Computacional. 2. Sistemas Inteligentes de Videovigilância. 3. Detecção de Pessoas. I. Brasileiro, Francisco Vilar. II. Gomes, Herman Martins. III. Título.

CDU 004.932(043)

MARCELL MANFRIN BARBACENA

**IMPACTO DA REDUÇÃO DA TAXA DE TRANSMISSÃO DE FLUXOS
DE VÍDEOS NA EFICÁCIA DE ALGORITMO PARA DETECÇÃO DE
PESSOAS**

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____

Prof. Dr. Francisco Vilar Brasileiro, UFCG
Doutor pela University of Newcastle upon Tyne, Newcastle upon Tyne,
Inglaterra.

Banca Examinadora:

Prof. Dr. Herman Martins Gomes, UFCG

Doutor pela University of Edinburgh, Edinburgh, Escócia

Prof. Dr. Robson Pequeno de Sousa, UEPB

Doutor pela Universidade Federal da Paraíba, Campina Grande, Brasil

Prof. Dr. Eanes Torres Pereira, UFCG

Doutor pela Universidade Federal de Campina Grande, Campina Grande, Brasil

Prof. Dr. Waslon Terllizzie Araújo Lopes, UFCG

Doutor pela Universidade Federal de Campina Grande, Campina Grande, Brasil

Coordenador do PPGCC: _____

Prof. Dr. Franklin de Souza Ramalho

Campina Grande, 2014.

Dedico este trabalho à minha esposa Alcyra Cotta por sua generosidade e apoio demonstrados em todos os momentos da vida que estamos construindo juntos. Aos meus filhos Miguel e Frida que alegam meu coração pelo simples fato de existirem.

Aos meus pais Ilton e Mazélia pelo carinho e ensinamentos e à minha amada irmã Marcella.

AGRADECIMENTOS

Agradeço a todos que de algum modo contribuíram para a realização deste trabalho, especialmente, aos meus pais, Ilton e Mazélia, pelos incentivos, bons conselhos e amorosidade sempre presente.

À minha irmã Marcellinha por sua alegria e ânimo contagiantes.

À minha amada esposa Alcyra.

Aos pequenos Miguel e Frida, luzes do meu viver.

Aos amigos que torceram e se felicitam pelo êxito do presente trabalho.

Agradeço aos meus prezados orientadores, Professor Francisco Vilar Brasileiro (Fubica) e Professor Herman Martins Gomes pela constante generosidade, paciência e apoio demonstrado ao longo da construção desta Dissertação.

Sobretudo, agradeço a Deus pelo dom da vida e do pensar.

RESUMO

Impulsionadas pela crescente demanda por sistemas de segurança para proteção do indivíduo e da propriedade nos dias atuais, várias pesquisas têm sido desenvolvidas com foco na implantação de sistemas de vigilância por vídeo com ampla cobertura. Um dos problemas de pesquisa em aberto nas áreas de visão computacional e redes de computadores envolvem a escalabilidade desses sistemas, principalmente devido ao aumento do número de câmeras transmitindo vídeos em tempo real para monitoramento e processamento. Neste contexto, o objetivo geral deste trabalho é avaliar o impacto que a redução da taxa de transmissão dos fluxos de vídeos impõe na eficácia dos algoritmos de detecção de pessoas utilizados em sistemas inteligentes de videovigilância. Foram realizados experimentos utilizando vídeos em alta resolução no contexto de vigilância com tomadas externas e com um algoritmo de detecção de pessoas baseado em histogramas de gradientes orientados, nos quais se coletou, como medida de eficácia do algoritmo, a métrica de área sob a curva de precisão e revocação para, em sequência, serem aplicados os testes estatísticos de Friedman e de comparações múltiplas com um controle na aferição das hipóteses levantadas. Os resultados obtidos indicaram que é possível uma redução da taxa de transmissão em mais de 70% sem que haja redução da eficácia do algoritmo de detecção de pessoas.

Palavras-chave: Visão computacional. Sistemas inteligentes de videovigilância. Detecção de pessoas.

ABSTRACT

Motivated by the growing demand for security systems to protect persons and properties in the nowadays, several researches have been developed focusing on the deployment of wide-area video coverage surveillance systems. One open research problem in the areas of computer vision and computer networks involves the scalability of these systems, mainly due to the increasing number of cameras transmitting real-time video for monitoring and processing. In this context, the aim of this study was to evaluate the impact that transmission data-rate reduction of video streams imposes on the effectiveness of people detection algorithms used in intelligent video surveillance systems. With a proposed experimental design, experiments were performed using high-resolution wide-area external coverage video surveillance and using an algorithm for people detection based on histograms of oriented gradients. As a measure of effectiveness of the people detection algorithm, the metric of area under the precision-recall curve was collected and statistical tests of Friedman and multiple comparisons with a control were applied to evaluate the hypotheses. The results indicated that it is possible to reduce transmission rate by more than 70% without decrease in the effectiveness of the people detection algorithm.

Keywords: Computer vision. Intelligent surveillance systems. People detection.

LISTA DE ILUSTRAÇÕES

Figura 1: Estágios de processamento de sistemas inteligentes de videovigilância.....	14
Figura 2: Exemplo de um sistema típico de videovigilância.....	19
Figura 3: Funcionamento simplificado do método HOG.....	21
Figura 4: Objeto em quadro de fluxo de vídeo.....	25
Figura 5: Imagens utilizadas no treinamento do detector HOG de DALAL (2005).....	29
Figura 6: Diagrama de atividades da execução dos experimentos.....	30
Figura 7: Detecção de pessoa no canto esquerdo da imagem.....	32
Figura 8: Detecção de pessoa no canto superior esquerdo da imagem em cena.....	32
Figura 9: Distribuição dos objetos nos fluxos de vídeos por tamanho vertical em pixels.	33
Figura 10: Taxa de transmissão dos fluxos de vídeos da amostra definitiva.....	34
Figura 11: Eficácia da detecção na hipótese 1 sob diferentes níveis de compressão.....	38
Figura 12: Eficácia da detecção na hipótese 2 sob diferentes níveis de resoluções espaciais.....	40
Figura 13: Eficácia da detecção na hipótese 3 sob diferentes níveis de compressão e resolução.....	42
Figura 14: Eficácia da detecção sob diferentes níveis de compressão e resoluções espaciais.....	43
Figura 15: Exemplo de imagem no formato original com objeto destacado.....	43
Figura 16: Exemplo de imagem redimensionada com objeto destacado.....	44
Figura 17: Exemplo de imagem do cenário 4.....	44
Figura 18: Exemplo de imagem do cenário 5.....	45
Figura 19: Eficácia da detecção no cenário 4 sob diferentes níveis de compressão.....	45

LISTA DE TABELAS

Tabela 1: Eficácia da detecção na amostra proposta inicialmente.....	31
Tabela 2: Tratamentos por hipótese.....	34
Tabela 3: Teste de normalidade da amostra.....	35
Tabela 4: Hipótese para teste de comparações múltiplas com um controle.....	36
Tabela 5: Teste de Friedman para a hipótese 1.....	37
Tabela 6: Análise <i>post-hoc</i> de múltiplas comparações para a hipótese 1.....	38
Tabela 7: Teste de Friedman para a hipótese 2.....	39
Tabela 8: Análise <i>post-hoc</i> de múltiplas comparações para a hipótese 2.....	39
Tabela 9: Teste de Friedman para a hipótese 3.....	41
Tabela 10: Análise <i>post-hoc</i> de múltiplas comparações para a hipótese 3.....	41

LISTA DE CÓDIGOS FONTES

Código fonte 1: Detector de objetos HOG.....	63
Código fonte 2: Cálculo da métrica de eficácia.....	65
Código fonte 3: Cálculo estatístico.....	69

LISTA DE SIGLAS E ABREVIATURAS

AMI:	Amazon Machine Image.
AUC-PR:	<i>Area under curve precision-recall</i> ou área sob a curva de precisão e revocação.
CDT:	C/C++ Development Tool.
EMR ou Amazon EMR:	Amazon Elastic MapReduce.
FHD:	<i>Full HD</i> .
HD:	<i>High Definition</i> ou alta definição.
hFHD:	<i>Half Full HD</i> .
hHD:	<i>Half HD</i> .
HOG:	<i>Histogram of Oriented Gradients</i> ou histograma de gradientes orientados.
H.264:	padrão para compressão de vídeo, baseado no MPEG-4 Part 10 ou AVC (<i>Advanced Video Coding</i>).
ICT:	<i>Information and communications technology</i> ou Infraestrutura de informação e comunicação.
IoT:	<i>Internet of Things</i> ou Internet das coisas.
Kbps:	kilobits por Segundo.
OpenCV:	Open Source Computer Vision.
PSNR:	<i>Peak Signal to Noise Ratio</i> ou Relação Sinal-Ruído de Pico.
QDA:	<i>Quadratic Discriminant Analysis</i> ou Análise Discriminante Quadrática.
ROI:	<i>region of interest</i> ou região de interesse.
SSIM:	Frame Rate Structural SIMilarity.
SVM:	<i>Support Vector Machine</i> ou Máquina de Vetores de Suporte.
TRECVID:	TREC Video Retrieval Evaluation.
UHD:	<i>Ultra HD</i> .
VIRAT:	Video and Image Retrieval and Analysis Tool.
VOC:	[Pascal] Visual Object Classes.

SUMÁRIO

1 INTRODUÇÃO.....	14
1.1 OBJETIVO.....	15
1.2 MOTIVAÇÃO E RELEVÂNCIA.....	16
1.3 ESTRUTURA DA DISSERTAÇÃO.....	16
2 REVISÃO DA LITERATURA.....	18
2.1 SISTEMAS INTELIGENTES DE VIGILÂNCIA.....	18
2.2 DETECÇÃO AUTOMÁTICA DE PESSOAS EM VÍDEO DIGITAL.....	20
2.2.1 Detecção de pessoas por subtração de imagem de fundo ou segmentação.....	20
2.2.2 Detecção direta de pessoas.....	21
2.2.3 Trabalhos relacionados.....	22
2.3 CONSIDERAÇÕES FINAIS.....	23
3 METODOLOGIA DE PESQUISA.....	25
3.1 HIPÓTESES DA PESQUISA.....	25
3.1.1 Fluxo de vídeo.....	25
3.1.2 Objeto em fluxo de vídeo.....	26
3.1.3 Algoritmo de detecção de objetos em quadro de vídeo e objetos representativos.....	27
3.1.4 Eficácia do algoritmo de detecção de objeto em quadro de vídeo.....	27
3.1.5 Eficácia do algoritmo de detecção de objetos em fluxo de vídeo.....	28
3.1.6 Hipóteses.....	28
3.2 ARQUITETURA EXPERIMENTAL.....	29
3.2.1 Execução dos experimentos.....	25
3.2.2 Amostragem.....	26
3.2.3 Tratamentos.....	27
3.2.4 Testes de hipóteses.....	27
3.3 CONSIDERAÇÕES FINAIS.....	35
4 EXPERIMENTOS E RESULTADOS.....	36
4.1 EXPERIMENTOS.....	36
4.1.1 Hipótese da compressão do fluxo de vídeo.....	36

4.1.2 Hipótese da redução da resolução espacial.....	39
4.1.3 Hipótese conjunta da compressão e redução da resolução espacial.....	41
4.2 AVALIAÇÕES ADICIONAIS.....	43
4.3 CONSIDERAÇÕES FINAIS.....	47
5 CONCLUSÃO.....	48
5.1 LIMITAÇÕES.....	49
5.2 TRABALHOS FUTUROS.....	49
REFERÊNCIAS.....	51
APÊNDICE A: MAPEAMENTO DE ARQUIVOS DE VÍDEO E NUMERAÇÕES.....	55
APÊNDICE B: RESULTADO DOS ENSAIOS DAS HIPÓTESES.....	56
APÊNDICE C: CÓDIGOS FONTES.....	63

1 INTRODUÇÃO

Os avanços técnicos na área de vídeo digital melhoraram não só a qualidade das imagens produzidas como baratearam o custo dos equipamentos de geração de vídeo permitindo uma popularização nas soluções de segurança eletrônica através de monitoramento por vídeo, os sistemas de videovigilância (WANG, 2013). De acordo com RÄTY (2010), o objetivo principal da instalação de sistemas de videovigilância em áreas públicas é reduzir atuações criminosas e terroristas. Furtos e roubos podem, muitas vezes, ser evitados apenas pela presença desses sistemas, que são utilizados para a solução desses casos através da análise forense dos vídeos. No entanto, eventos mais severos, como ataques terroristas ou danos ao patrimônio, exigem um monitoramento contínuo dos vídeos por operadores humanos e ação de agentes de segurança no local dos eventos. A pesquisa em monitoramento inteligente por vídeo visa exatamente auxiliar no monitoramento contínuo dos vídeos, sendo hoje, uma das mais ativas na área de visão computacional (HU et al., 2004; JI et al., 2010).

Os sistemas de videovigilância com monitoramento contínuo por operadores humanos podem ser classificados como proativos e reativos. No monitoramento proativo, o operador fica continuamente observando os vídeos de forma a detectar e prevenir incidentes. Nesse tipo de monitoramento, de acordo com DADASHI (2008), após 12 minutos, 45% das informações podem ser perdidas, e após 22 minutos pode-se chegar a 95%. No monitoramento reativo, os operadores são alertados por algum evento, por exemplo, a informação de comportamento suspeito em determinada localização, sendo então utilizados os vídeos para o monitoramento contínuo do local do evento para auxiliar os agentes de segurança que atuarão no local do evento. Os sistemas inteligentes de videovigilância são projetados exatamente para auxiliar operadores no monitoramento proativo contínuo podendo os operadores mudar para o modo reativo após serem alertados por eventos (automáticos ou não) (CHATTOPADHYAYR et al., 2013; POPOOLA; WANG, 2012).

Os principais estágios de processamento de sistemas inteligentes de videovigilância são: detecção e reconhecimento de objetos, rastreamento, análise de comportamento e atividade, e armazenamento (VALERA; VELASTIN, 2005). A Figura 1 ilustra esses estágios e, por ser o estágio inicial, a pesquisa em detecção de pessoas em imagens é um dos problemas principais para sistemas inteligentes de videovigilância, assim como para outras aplicações, por exemplo, direção autônoma e robótica (ENZWEILER; GAVRILA, 2009).

Outro problema de pesquisa comum em sistemas inteligentes de videovigilância é a escalabilidade da solução na transmissão dos fluxos de vídeos, principalmente com a

tendência da utilização desses sistemas no contexto de cidades inteligentes (CHATTOPADHYAYR et al., 2013). Para se ter uma ideia da problemática, em cerca de 15 anos, o número de câmeras de vigilância no Reino Unido passou de 5.238 (cinco mil duzentos e trinta oito) para aproximadamente 5.900.000 (cinco milhões e novecentos mil) (DADASHI, 2008; REEVE, 2013).

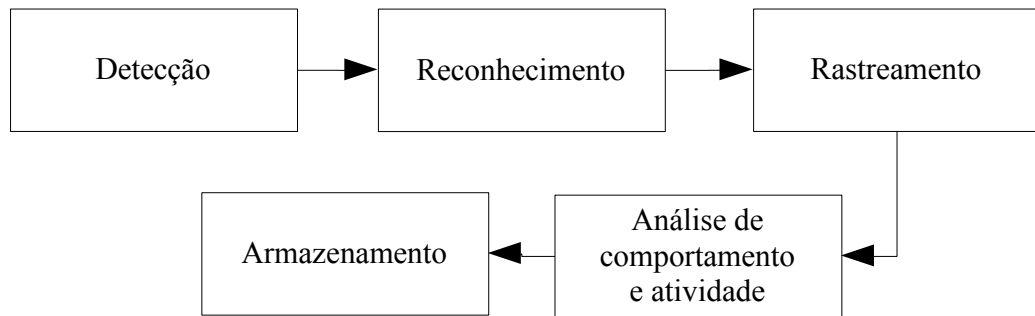


Figura 1: Estágios de processamento de sistemas inteligentes de videovigilância. Adaptado de (VALERA; VELASTIN, 2005).

1.1 OBJETIVO

O objetivo geral deste trabalho é avaliar o impacto da redução da taxa de transmissão dos fluxos de vídeos, utilizados em sistemas inteligentes de videovigilância, na eficácia da detecção de pessoas dos algoritmos utilizados nesses sistemas. A eficácia dos algoritmos na detecção de pessoas foi avaliada por intermédio da métrica de área sob a curva de precisão e revocação (*recall*) em cada quadro do fluxo de vídeo. Tais algoritmos utilizam apenas um fluxo de vídeo para sua execução, não sendo considerada para esta pesquisa a detecção entre diferentes fluxos de vídeo.

Os fluxos de vídeo utilizados nesta pesquisa são provenientes de câmeras de alta definição, estacionárias, passivas, monoculares, coloridas, em tomadas externas com ampla cobertura e com eventos sem atuação direta de atores, ou com atuação minimizada, permitindo o máximo de correspondência com vídeos reais de sistemas de videovigilância.

A redução da taxa de transmissão de um fluxo de vídeo foi realizada das seguintes formas: (1) por meio da utilização de algoritmos de compactação de vídeo com perdas, utilizando o codificador H.264; (2) por meio da redução da quantidade de pontos (*pixels*) de cada quadro do fluxo via redimensionamentos da resolução espacial do vídeo mantendo-se, entretanto, a razão de aspecto (*aspect ratio*) do vídeo.

1.2 MOTIVAÇÃO E RELEVÂNCIA

Em 2015, espera-se que o mercado de vigilância atinja mais de 40 bilhões de dólares (MARKETRESEARCH.COM, 2011) devido ao aumento no interesse da sociedade em assuntos relativos à proteção e segurança de seus indivíduos e patrimônio, resultando em vários experimentos relacionados com a implantação de sistemas de vigilância por vídeo com alta cobertura espacial (GIRGENSOHN et al., 2007; JAVED et al., 2003; SONG; TAO; MAYBANK, 2013). A integração desses sistemas com os avanços na área de infraestrutura de informação e comunicação (ICT) no contexto de cidades inteligentes (*Smart Cities*) possibilitará uma ampliação da cobertura espacial e na habilidade de detectar e identificar situações de interesse para esses sistemas (CALAVIA et al., 2012).

No entanto, a implantação de sistemas de vigilância por vídeo utilizando Internet das coisas (IoT) (GUBBI et al., 2013), ou outra infraestrutura de informação e comunicação de cidades inteligentes, aumenta o problema de escala da implantação desses sistemas, uma vez que a utilização da largura de banda da infraestrutura para transmissão simultânea de vários fluxos de vídeo pode provocar uma degradação do desempenho de outros serviços disponibilizados (HERNÁNDEZ-MUÑOZ; VERCHER; MUÑOZ, 2011; KORSHUNOV; OOI, 2005).

Nesse sentido, encontrar uma melhor utilização da largura de banda da rede de transmissão para sistemas de videovigilância poderá não só aumentar a capacidade desses sistemas, v.g., por meio do aumento no número de câmeras do sistema, como também facilitar a adoção de infraestruturas de cidades inteligentes para a implantação desses sistemas no contexto de segurança pública.

Esta pesquisa propõe verificar se é possível aperfeiçoar a transmissão dos fluxos de vídeos em sistemas de videovigilância, permitindo que a infraestrutura de comunicação utilizada coexista com outros serviços disponibilizados.

1.3 ESTRUTURA DA DISSERTAÇÃO

Esta dissertação está organizada da seguinte forma: o Capítulo 2 apresenta a revisão de literatura, que começa com a evolução dos sistemas de segurança, em especial os sistemas inteligentes de videovigilância. Em seguida, são apresentados conceitos de detecção de objetos com um detalhamento do algoritmo *Histogram of Oriented Gradients* (HOG).

Finalmente apresenta-se uma revisão bibliográfica de trabalhos anteriores relacionados à pesquisa.

No Capítulo 3, discute-se a metodologia utilizada nesta pesquisa assim como são detalhadas as ferramentas e configurações utilizadas para a realização dos experimentos.

Em seguida, no Capítulo 4, apresentam-se os resultados dos experimentos e a análise das hipóteses levantadas nesta pesquisa.

Finalmente, as conclusões da pesquisa, suas limitações e trabalhos futuros são discutidas no Capítulo 5.

2 REVISÃO DA LITERATURA

Neste capítulo, é apresentada a revisão da literatura utilizada para o embasamento teórico-metodológico do presente estudo, assim como a definição dos conceitos utilizados. A partir da análise da bibliografia busca-se identificar o estado-da-arte do tema em análise, relatando não apenas os avanços científicos alcançados acerca do tema, mas também seus entraves e suas limitações teóricas atuais.

A primeira seção deste capítulo apresenta os conceitos de sistemas inteligentes de vigilância, sua evolução tecnológica e principais problemas relacionados. Na seção seguinte, são discutidas as técnicas para detecção de objetos em fluxos de vídeos, assim como os avanços obtidos e futuras perspectivas na área quando o objetivo é a detecção de pessoas. Na última seção, serão apresentadas pesquisas que tem relacionamento mais próximo aos objetivos específicos deste trabalho, ou seja, relacionam a qualidade de fluxos de vídeos com a eficiência de algoritmos de detecção de objetos.

2.1 SISTEMAS INTELIGENTES DE VIGILÂNCIA

O propósito principal dos sistemas inteligentes de vigilância é mostrar e alertar seus operadores quando da ocorrência de uma situação extraordinária. Em outras palavras, o sistema interpreta cenas em tempo real auxiliando os operadores na tomada de decisão (AHMEDALI; CLARK, 2006)

A utilização de sistemas inteligentes de vigilância é praticamente inevitável em situações quando há grande quantidade de câmeras impossibilitando o monitoramento proativo, ou quando a tarefa é difícil ou desagradável para os operadores como na busca por objetos abandonados, ou quando há a necessidade de tomada de ações de forma mais imediata como, por exemplo, no monitoramento de eventos de grande porte (DADASHI, 2008).

Segundo RÄTY (2010), os sistemas de vigilância podem ser divididos em 3 gerações. A primeira geração é caracterizada pela utilização de equipamentos analógicos em todo o sistema, funcionando basicamente como uma extensão perceptiva do operador. A segunda geração é caracterizada pela introdução de equipamentos digitais permitindo alguma automação como a detecção de eventos e alarmes, entretanto ainda não é possível a realização de automações mais complexas como rastreamentos e análise comportamental. A terceira geração é caracterizada pela transição completa para as tecnologias digitais e utilização desses sistemas em largas áreas e com um número superior de sensores. Uma segunda característica

dos sistemas de terceira geração é que a inteligência do sistema pode funcionar de forma distribuída por meio da introdução de câmeras inteligentes, redes de sensores sem fio e robôs autônomos.

Na área de visão computacional, destacam-se as pesquisas envolvendo a melhoria dos algoritmos de detecção, rastreamento e reconhecimento de pessoas. As compilações de POPOOLA (2012) e HU et al. (2004) nesse assunto mostram que essa área obteve avanços expressivos na incorporação desses algoritmos para a melhoria da inteligência em sistemas de videovigilância, entretanto o desempenho desses sistemas ainda não alcançou os requisitos mínimos para um monitoramento em tempo real (DADASHI, 2008).

Outro problema de pesquisa em aberto envolve a escalabilidade desses sistemas (KORSHUNOV; OOI, 2005; RÄTY, 2010). Na tentativa de evitar esses problemas de escalabilidade, os sistemas KNIGHT (JAVED et al., 2003) e SfinX (DIRNITRIJEVIC; CHANG, 2003) transmitem os fluxos de vídeos utilizando parâmetros fixos de compactação para um local centralizado onde se realiza o processamento. Por sua vez o sistema DETER (PAVLIDIS et al., 2001) pressupõe a utilização de uma infraestrutura de transmissão dedicada e os sistemas propostos por NAIR (2002) e (YUAN et al. (2003) utilizam o envio de imagens estáticas para evitar a utilização excessiva da infraestrutura. Em geral, os problemas de escalabilidade ocorrem principalmente quando são sistemas de videovigilância que transmitem fluxos de vídeo em tempo real para uma localidade central onde ocorre o monitoramento e processamento.



Figura 2: **Exemplo de um sistema típico de videovigilância.** Adaptado de (TEAM, 2000c).

Na Figura 2 observa-se um exemplo de sistema de videovigilância onde os fluxos de vídeo são enviados para uma localidade central (canto inferior direito da figura) onde pode ser monitorado por operador humanos (sala de controle ou *cockpit*) ou automaticamente. As câmeras do lado esquerdo da figura enviam seus fluxos de vídeos para uma localidade próxima centralizada onde pode haver algum tipo de processamento como por exemplo a gravação local dos fluxos de vídeos.

2.2 DETECÇÃO AUTOMÁTICA DE PESSOAS EM VÍDEO DIGITAL

Detectar pessoas em vídeo digital tem sido uma das áreas com extenso interesse nos últimos anos, sendo muitas técnicas e modelos propostos e aplicados. Uma demonstração desse interesse foi que a partir de 2005 se iniciou a competição *Pascal Visual Object Classes (VOC)* em que a comunidade de visão computacional avalia anualmente as melhorias nos algoritmos de detecção (EVERINGHAM et al., 2009).

As dificuldades inerentes das características humanas em vídeo, como variações de posicionamento e oclusão, fazem com que diferentes métodos e técnicas para a detecção tenham sido propostas, sendo que a literatura relevante divide-se entre técnicas que requerem um pré-processamento das imagens (subtração de imagens de fundo ou uso de segmentação em contraste com detecção direta), as que utilizam características que descrevem a aparência humana (baseadas no formato, cor ou movimento) e as técnicas que utilizam modelos explícitos do corpo humano aliadas com aprendizado automatizado (baseadas na silhueta do corpo) (TIAN et al., 2011).

Nas subseções seguintes, são apresentados maiores detalhes sobre uma amostra representativa dos métodos e técnicas para detecção automática de pessoas em vídeo digital, desde as mais simples, como a técnica de subtração de fundo ou segmentação, até técnicas mais sofisticadas como o histograma de gradiente orientado.

2.2.1 Detecção de pessoas por subtração de imagem de fundo ou segmentação

Técnicas baseadas em subtração de imagem de fundo buscam isolar objetos em primeiro plano para então classificar esses objetos encontrados em categorias como pessoas, animais, veículos, etc., baseadas no formato, cor, movimento ou outras características (JOSHI; PANCHAL, 2014).

O principal conceito dessa abordagem é a utilização das diferenças entre quadros consecutivos do fluxo de vídeo para a detecção de objetos em movimento. O fundo da imagem então é caracterizado pelo cenário sem os objetos em movimento. Os problemas dessa abordagem são relacionados à variação de iluminação, efeitos fantasmas, mas principalmente à impossibilidade de detectar pessoas que não estão em movimento (PAUL; HAQUE; CHAKRABORTY, 2013).

Os métodos de destaque com essa abordagem são os que utilizam misturas gaussianas adaptativas (TIAN et al., 2011), imagem não paramétrica de fundo (LANZA; SALTI; STEFANO, 2011) e imagem hierárquica de fundo (CHEN et al., 2012).

2.2.2 Detecção direta de pessoas

Os métodos de detecção direta são geralmente baseados na extração de contornos de objetos para compará-los com modelos pré-definidos de pessoas. Esses métodos podem ser subdivididos em dois tipos: os que têm uma abordagem holística, em que o modelo pré-definido é do corpo inteiro; e os que têm uma abordagem baseada em partes (FELZENSZWALB et al., 2010), nos quais os resultados da detecção de cada parte do objeto são combinados por meio de um esquema de votação probabilística, gramática ou outro modelo.

Um dos métodos mais populares nesta abordagem é a utilização de Histograma de Gradientes Orientados (HOG) combinados com classificadores baseados em Máquina de Vetores de Suporte Lineares (SVM) (ADAMS; FERRYMAN, 2013). Esse método utiliza um dos conceitos fundamentais na área de visão computacional: gradiente primário ou gradiente orientado de imagem. Devido à simplicidade, usualmente aplica-se o gradiente primário em todos os *pixels* da imagem alvo (GONZALEZ; WOODS, 2000). O gradiente f na coordenada (x, y) , é definido como o vetor:

$$f(x, y) = \nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} |(x-1, y) - (x+1, y)| \\ |(x, y-1) - (x, y+1)| \end{bmatrix} \quad (1)$$

As informações de magnitude e direção desse vetor são utilizadas no método HOG e definidas como:

$$\text{magnitude}(\nabla f) = [G_x^2 + G_y^2]^{1/2} \quad (2)$$

$$\text{direção}(\nabla f) = \tan^{-1} \left(\frac{G_x}{G_y} \right) \quad (3)$$

O detector de pessoas HOG proposto em DALAL (2005) utiliza uma janela de detecção de 64×128 pixels e a base de imagens MIT e INRIA para treinamento e verificação experimental. A janela de detecção é dividida em 7×15 blocos, num total de 105 blocos, onde cada bloco possui 2×2 células de 8×8 pixels. Em cada célula é computado um histograma de 9 células (*bins*) com os valores de ângulo e magnitude dos gradientes orientados de cada pixel. Para melhorar a invariância com relação à iluminação, os valores de histogramas de cada célula dos blocos da janela de detecção são normalizados. O resultado desse último passo é conhecido como descritor HOG. Na detecção, é realizada uma busca com a janela de detecção em toda a imagem utilizando um classificador baseado em máquina de vetores de suporte (*Support Vector Machine* ou SVM) alimentado com os descritores HOG calculados.

As principais limitações desse método são a utilização de modelos em 2-D para a representação do objeto a ser detectado, exigindo por isso uma base de treinamento com grande representatividade de variações do modelo a ser detectado e devido à computação do descritor HOG ser computacionalmente intensiva, uma implementação em tempo real do método ainda é um problema em aberto. A Figura 3 ilustra as etapas de funcionamento do método HOG.

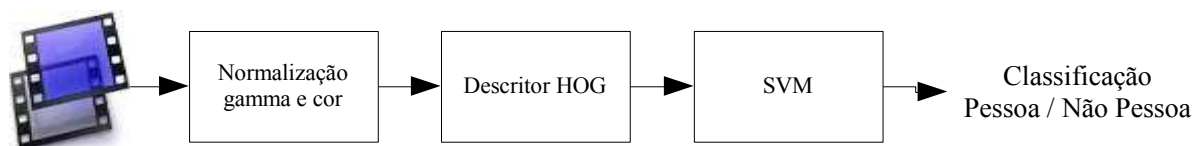


Figura 3: **Funcionamento simplificado do método HOG** adaptado de DALAL (2005).

2.2.3 Trabalhos relacionados

A demanda por sistemas de videovigilância fez com que vários sistemas comerciais fossem desenvolvidos nos últimos anos. No entanto, VALERA (2005) analisa alguns desses sistemas inteligentes de videovigilância e aponta dificuldade no desenvolvimento de um sistema inteligente com uma grande cobertura espacial. RÄTY (2010) indica que o problema da escalabilidade seria o principal obstáculo nesse tipo de sistema.

Nesse sentido, uma das abordagens mais promissoras para mitigar o problema de escalabilidade das transmissões dos fluxos de vídeos em sistemas de videovigilância é a utilização de compactação dos fluxos de vídeos (DIRNITRIJEVIC; CHANG, 2003; JAVED et al., 2003). No entanto, a compactação implica na redução da qualidade do fluxo de vídeo,

pois não poderia degradar ao ponto de comprometer a eficácia dos algoritmos de inteligência computacional utilizados, como por exemplo, os algoritmos de detecção de objetos.

As pesquisas realizadas em KORSHUNOV (2005, 2011) propõem definições de qualidade crítica, mínima em fluxos de vídeo para que algoritmos de detecção, reconhecimento e rastreamento de faces humanas sejam utilizados. Os autores utilizam 3 detectores de faces – o detector Viola-Jones (VIOLA; JONES, 2004), o detector Rowley (ROWLEY; BALUJA; KANADE, 1998) e um detector baseado em QDA – e 3 métodos de compactação - por meio da compressão JPEG, do redimensionamento espacial, e da retirada de quadros dos fluxos de vídeos. A eficácia dos detectores de faces humanas é avaliada quando aplicadas às estratégias de compactação, para isso os experimentos utilizam 333 imagens das bases MIT/CMU e Yale, e 2 fluxos de vídeos elaborados pelos autores, com resolução espacial máxima de 352 x 288 *pixels*. Os resultados nos fluxos de vídeos indicam quase nenhuma perda de eficácia até que cheguem a um nível crítico de compactação.

O estudo por KAFETZAKIS et al. (2013) analisa o impacto da qualidade dos fluxos de vídeos em algoritmos de detecção de pessoas, baseados no HOG, e de movimento, baseados no detector Harris Corner (HARRIS; STEPHENS, 1988) e do algoritmo de rastreamento de Kanade-Lucas-Tomasi (TOMASI; KANADE, 1991). A pesquisa utiliza as métricas de qualidade de vídeo, como a *relação sinal-ruído de pico* (PSNR) (ESKICIOGLU; FISHER, 1995) e *Frame Rate Structural SIMilarity* (SSIM) (WANG et al., 2004), na avaliação da compactação dos fluxos de vídeo e nos experimentos utilizam 3 fluxos de vídeos da base TRECVID (SMEATON; OVER; KRAAIJ, 2006). Todos os fluxos de vídeos possuíam originalmente as mesmas características de compactação variável, com aproximadamente 6 mbps, e resolução espacial de 720 x 576. Os resultados no detector de pessoas HOG evidenciam uma redução na eficácia do detector tanto no aumento da compactação dos fluxos de vídeos, quanto na redução da resolução espacial.

Muito embora os sistemas inteligentes de videovigilância sejam diretamente afetados pela codificação e compactação dos fluxos de vídeos existem poucas investigações sobre o impacto disso nos algoritmos de detecção. Este trabalho diferencia-se dos anteriores principalmente pela utilização de uma abordagem mais formal na análise dos resultados da eficácia do algoritmo de detecção de pessoas através do uso de uma metodologia experimental com avaliação estatística dos resultados.

2.3 CONSIDERAÇÕES FINAIS

O propósito principal deste capítulo foi apresentar uma contextualização e o estado-da-arte do tema em análise, iniciando com a apresentação dos conceitos de sistemas inteligentes de vigilância, sua evolução tecnológica e principais problemas relacionados. Como também, foram discutidas as técnicas para detecção de objetos em fluxos de vídeos, bem como os avanços obtidos e as futuras perspectivas teóricas na área quando o objetivo é a detecção de pessoas. Por fim, foram apresentadas pesquisas que relacionam a qualidade de fluxos de vídeos com a eficiência de algoritmos de detecção de objetos, trabalhos estes que tem relacionamento próximo aos objetivos específicos desta pesquisa.

No próximo capítulo, serão formalizados os objetivos, hipóteses e a metodologia adotado neste trabalho, utilizando os conceitos apresentados neste capítulo.

3 METODOLOGIA DE PESQUISA

Este capítulo discute os procedimentos metodológicos que norteiam o objeto de estudo em questão. Inicialmente é apresentada a definição das hipóteses desenvolvidas, e, em seguida, o método experimental juntamente com a seleção amostral adotada. Por fim, o modelo estatístico e as ferramentas utilizadas para validar as hipóteses desenvolvidas.

3.1 HIPÓTESES DA PESQUISA

Preliminarmente, faz-se necessário desenvolver alguns conceitos necessários para a definição das hipóteses.

3.1.1 Fluxo de vídeo

Um fluxo de vídeo V pode ser caracterizado por um conjunto enumerado Q de quadros de vídeo e variáveis que descrevem as características do vídeo como o fator de compressão c e a resolução espacial r .

$$V = \langle Q, c, r \rangle \quad (4)$$

$$Q = \{q_1, \dots, q_n, \dots, q_f\} \quad (5)$$

em que f é quantidade de quadros do vídeo e c é expresso em *kbps* (kilobits por segundo).

$$r = \langle h, w \rangle \quad (6)$$

em que h é o tamanho vertical e w o tamanho horizontal da resolução espacial do fluxo de vídeo.

Algumas resoluções espaciais são padronizadas (SMPTE, 2001, 2003), como:

- *Full HD* (FHD): possui tamanho horizontal de 1920 e vertical de 1080, ou seja, 1920 x 1080 *pixels*.
- *HD*: possui tamanho horizontal de 1280 e vertical de 720, ou seja, 1280 x 720 *pixels*.

Neste trabalho, faremos referência a mais duas resoluções:

- *Half Full HD* (hFHD): exatamente metade dos valores da resolução FHD, ou seja 960 x 540 *pixels*.
- *Half HD* (hHD): exatamente metade dos valores da resolução HD, ou seja 640 x 360 *pixels*.

3.1.2 Objeto em fluxo de vídeo

Um mesmo objeto pode estar presente em vários quadros de um vídeo, podendo em cada quadro, apresentar características espaciais – configurações de *pixels* – que o diferenciam entre os quadros. Por exemplo, um carro estacionado estará presente no fluxo de vídeo em vários quadros com praticamente as mesmas características espaciais, entretanto uma pessoa que está em um cruzamento, atravessando a rua horizontal em direção à câmera e depois virando à esquina de forma a atravessar a rua vertical adjacente, terá sua representação espacial alterada a cada quadro (primeiramente vista pela frente, depois lateralmente, e finalmente por trás). Dessa forma, representaremos um objeto O no quadro q_n como um conjunto de coordenadas espaciais.

$$O_{q_n} = \langle (x, y), (z, w) \rangle \quad (7)$$

em que (x, y) são as coordenadas superiores esquerdo e (z, w) é as coordenadas inferiores direito dos *pixels* no quadro.

Pode-se então afirmar que o objeto O_{q_n} é um retângulo conforme exemplo da Figura 4.



Figura 4: Objeto em quadro de fluxo de vídeo

3.1.3 Algoritmo de detecção de objetos em quadro de vídeo e objetos representativos

Pode-se definir um algoritmo de detecção de objetos D aplicado em um quadro de vídeo q_n como uma função que tem como resultado um conjunto enumerado de objetos detectados.

$$D(q_n) = \{d_1, d_2, \dots, d_g\} \quad (8)$$

$$\forall d_m \in D(q_n), d_m = \langle (r, s), (t, u) \rangle \quad (9)$$

em que g é quantidade de objetos detectados no quadro, d_i representa um objeto detectado pelo algoritmo, (r, s) são as coordenadas superiores esquerdo e (t, u) são as coordenadas inferiores direito dos *pixels* no quadro do objeto detectado.

A ordem em que os objetos detectados ocorrem no conjunto representa a probabilidade dada pelo algoritmo, ou seja, o primeiro objeto do conjunto tem maior probabilidade de ocorrência que o segundo objeto.

$$T(q_n) = \{t_1, t_2, \dots, t_g\} \quad (10)$$

$$\forall d_x, d_y \in D(q_n) \text{ e } x < y, p(d_x) > p(d_y) \quad (11)$$

em que $T(q_n)$ são os objetos representativos no quadro q_n , aqueles realmente existentes no quadro, e a função $p(d)$ é a probabilidade dada pelo algoritmo de que o objeto detectado d pertença ao conjunto $T(q_n)$.

3.1.4 Eficácia do algoritmo de detecção de objeto em quadro de vídeo

A eficácia de algoritmo de detecção de objeto é similar à métrica da área sob a curva de precisão e revocação (AUC-PR) utilizada em recuperação de informação (ZHU, 2004). Para efetuar o julgamento se a detecção do objeto é um verdadeiro ou um falso positivo utiliza-se uma função ζ que mensura a sobreposição entre os retângulos do objeto presente no quadro do vídeo O_{q_n} e o objeto detectado d_m .

$$\zeta(O_{q_n}, d_m) = \begin{cases} \text{Verdadeiro,} & \text{caso } O_{q_n} \cap d_m > \delta \\ \text{Falso,} & \text{caso contrário} \end{cases} \quad (12)$$

$$AUC(q_n) = \sum_{i=1}^{|D(q_n)|} \text{precisao}(\zeta, D, i) \times \Delta \text{revocacao}(\zeta, D, i) \quad (13)$$

em que $\text{precisao}(\zeta, D, i)$ é o valor de precisão até o objeto d_i e $\Delta \text{revocacao}(\zeta, D, i)$ é a variação da revocação (*recall*) entre os objetos d_{i-1} e d_i . Os detalhes da formalização das funções $\text{precisao}()$ e $\Delta \text{revocacao}()$ podem ser obtidos em ZHU (2004).

A Equação 13 descreve a área sob a curva de precisão e revocação. Neste trabalho, adotou-se o valor 10% para δ na detecção de objetos da função ζ em contraposição ao padrão de 50% adotado no *Pascal Visual Object Classes* (VOC) (EVERINGHAM et al., 2009). Essa restrição mais fraca deve-se à proporção entre o tamanho dos objetos a serem detectados e a resolução do quadro do fluxo de vídeo ser inferior na base do VIRAT que a encontrada no VOC, ou seja, na base de vídeos VIRAT, os objetos são menores em relação ao tamanho do quadro. Neste contexto, mesmo que o detector aponte uma área de detecção

proporcionalmente superior para o objeto detectado, essa superioridade de área não representará uma proporção significativa da área total do quadro do vídeo.

3.1.5 Eficácia do algoritmo de detecção de objetos em fluxo de vídeo

A métrica de eficácia do algoritmo D é a métrica AUC-PR considerando todos os quadros do vídeo como se fosse um único quadro conforme Equação 14.

$$AUC(V) = AUC(\langle Q, c, r \rangle) = AUC\left(\bigcup_{i=1}^{|Q|} (q_i)\right) \quad (14)$$

3.1.6 Hipóteses

Conforme mencionado anteriormente, o objetivo deste trabalho é avaliar o impacto da redução da taxa de transmissão do fluxo de vídeo em algoritmos de detecção de pessoas aplicados ao fluxo de vídeo. Nesse sentido, o desempenho do algoritmo de detecção é avaliado quando se aplica variação de parâmetros que afetam direta e indiretamente a redução da taxa de transmissão de vídeo, quais sejam: o fator de compressão do fluxo de vídeo e a resolução espacial do vídeo.

As hipóteses formuladas para realizar este trabalho são:

1. Hipótese da compressão do fluxo de vídeo

Existe um fator máximo de compressão do fluxo de vídeo tal que, se superado, a eficácia do algoritmo de detecção diminui conforme Equação 15.

$$\exists (c_{min} < c) \mid AUC(\langle Q, c, r \rangle) = AUC(\langle Q, c_{min}, r \rangle) \quad (15)$$

2. Hipótese da redução da resolução espacial

Existe uma resolução espacial mínima para o fluxo de vídeo tal que, se reduzida, a eficácia do algoritmo de detecção diminui conforme Equação 16.

$$\exists (r_{min} < r) \mid AUC(\langle Q, c, r \rangle) = AUC(\langle Q, c, r_{min} \rangle) \quad (16)$$

3. Hipótese conjunta da compressão e redução da resolução espacial

Existe uma combinação de fator máximo de compressão e de resolução espacial mínima tal que, se o fator de compressão for aumentado ou a resolução espacial for reduzida, a eficácia do algoritmo de detecção diminui conforme Equação 17.

$$\exists \{(c_{min} < c) \text{ e } (r_{min} < r)\} \mid AUC(\langle Q, c, r \rangle) = AUC(\langle Q, c_{min}, r_{min} \rangle) \quad (17)$$

Considera-se para as hipóteses 2 e 3 que as alterações da resolução espacial dos fluxos de vídeos mantem a proporcionalidade vertical e horizontal do fluxo de vídeo.

3.2 ARQUITETURA EXPERIMENTAL

Neste trabalho se utiliza o método experimental na avaliação das hipóteses formuladas anteriormente. Com isso, foi analisado o impacto na eficácia de um algoritmo de detecção de pessoas quando reduzida a taxa de transmissão do fluxo de vídeo por meio do aumento do nível de compactação do codificador de vídeo H.264, assim como, reduzindo a quantidade de *pixels* por quadro através de um redimensionamento espacial do fluxo de vídeo.

Os vídeos selecionados para os experimentos foram um subconjunto da base VIRAT na versão 2.0 (OH et al., 2011) que continham ao menos um objeto da classe pessoa anotado e que estavam na resolução *Full HD*. A base de vídeos VIRAT foi selecionada por ter sido recentemente atualizada e possuir mais de 30 horas de vídeo em tomadas externas reais, com ações ocorrendo naturalmente sem o uso de atores.

Na escolha do algoritmo de detecção de pessoas foi adotada uma opção baseada em histogramas de gradientes orientados (HOG) treinados para a detecção de pessoas. Essa escolha deve-se ao fato de o seu desempenho computacional ser consideravelmente superior a outros métodos, bem como, por representar um dos grandes avanços na área de detectores, sendo hoje, uma das tecnologias mais utilizadas neste campo (ADAMS; FERRYMAN, 2013). Além disso, embora a qualidade dos vídeos seja alta, *i.e. Full HD*, as pessoas nos vídeos têm entre 20 e 180 *pixels* de altura, o que impõe grande dificuldade ao algoritmo de detecção.

A métrica coletada foi a eficácia do algoritmo HOG em fluxos de vídeos, ou seja, a área sob a curva de precisão e revocação considerando-se todos os quadros do vídeo da amostra como se fosse um único quadro. Definida a métrica a ser coletada, fez-se a seleção da amostra a partir dos fluxos de vídeos da base VIRAT. Em seguida, analisou-se a amostra para a definição dos tratamentos e testes estatísticos que foram utilizados. O detalhamento dessas atividades está descrito nas subseções a seguir.

3.2.1 Execução dos experimentos

Devido aos requisitos de desempenho das aplicações em visão computacional, o detector de objetos na pesquisa foi desenvolvido na linguagem C (ANSI, 1989), tendo como ambiente de desenvolvimento a plataforma Eclipse® (FOUNDATION, 2014) com o *plugin* CDT (*C Development Tool*).

O detector de objetos HOG utilizado foi o que está disponível na biblioteca OpenCV (BRADSKI, 2000), que foi desenvolvida originalmente pela Intel®, e disponibilizada em

código aberto. A configuração utilizada para o detector HOG foi para detecção de objetos com janela de tamanho 64×128 pixels, célula de tamanho 8×8 pixels e 5% de aumento na escala da janela de detecção. Ademais o detector de objetos foi configurado com o treinamento para a detecção de pessoas resultado do trabalho de DALAL (2005), que utilizou imagens de tamanho vertical de 128 pixels, entretanto as pessoas possuíam tamanho superior a 96 pixels conforme ilustrado nas imagens exemplo da Figura 5.



Figura 5: Imagens utilizadas no treinamento do detector HOG de (DALAL; TRIGGS, 2005).

Os fluxos de vídeos foram compactados e convertidos para produção dos níveis da amostra utilizando a biblioteca FFmpeg (TEAM, 2000a), que é de código aberto e que permite a codificação e decodificação de arquivos, fluxos de vídeo e imagens. Essa biblioteca serve de base para diversas aplicações como MPlayer (TEAM, 2000b), xine (TEAM, 2000c) e VLC (VIDEOLAN, 2001).

A execução do detector de objetos em cada fluxo de vídeo utilizou a plataforma Amazon EMR (AMAZON, 2009) em um *cluster* com até 20 máquinas virtuais do tipo *c1.medium* e configurada com a *Amazon Machine Image* (AMI) na versão 3.0.4, sendo cada fluxo de vídeo processado paralelamente otimizando a execução dos experimentos.

Para a coleta da métrica de eficácia do detector de objetos no fluxo de vídeos foi desenvolvida uma aplicação em Python (ROSSUM; DRAKE, 2001) que após cada execução do detector de objetos em um fluxo vídeo produzia o valor da métrica AUC-PR considerando todos os quadros do vídeo como se um único quadro através da comparação dos resultados do detector de objetos e das anotações originais dos objetos disponíveis na base VIRAT. Para a execução dos testes estatísticos foi utilizada a linguagem R (R CORE TEAM, 2014).

O diagrama da Figura 6 ilustra a interligação dessas ferramentas e como se deu a execução dos experimentos.

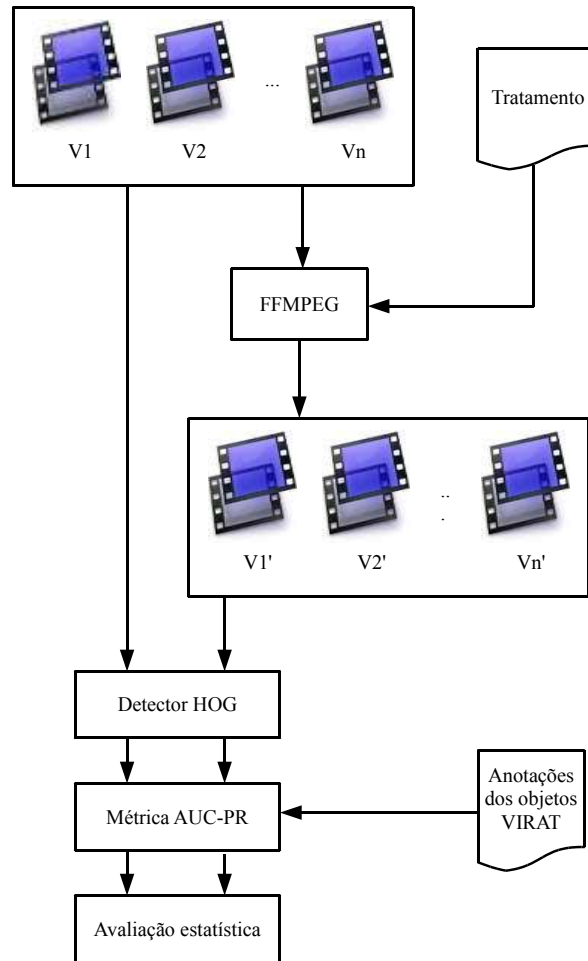


Figura 6: Diagrama de atividades da execução dos experimentos.

3.2.2 Amostragem

A priori, a proposta de amostragem inicial utilizou todos os vídeos disponíveis na base de vídeos VIRAT que estivessem em resolução *Full HD* e contivessem pelo menos uma anotação de objeto do tipo pessoa. Com essa definição a amostra inicial possuiu 101 fluxos de vídeos produzidos por 7 câmeras, que denominamos cenários conforme recomendação da base de vídeos VIRAT. Os resultados da métrica de eficácia do detector nesses fluxos de vídeos podem ser vistos na Tabela 1.

Conforme pode ser observado na Tabela 1, em vários fluxos de vídeos, a eficácia do detector se mostrou demasiadamente baixa, algumas vezes nula. Os motivos para isso vão desde a poucos objetos a serem detectados no fluxo de vídeo até as condições ambientais em que o fluxo de vídeo foi gravado (com chuva, escuro, etc.), conforme ilustrado nas Figuras 7 e 8.

Tabela 1: Eficácia da detecção na amostra proposta inicialmente. Em vermelho, os fluxos de vídeos eliminados da amostra inicial.

Vídeos	Cenários						
	1	2	3	4	5	6	7
1	75,3	95,1	0,0	6,3	28,2	0,0	2,8
2	84,2	96,0	0,0	16,9	42,9	10,5	8,1
3	5,6		0,0	10,3	25,0	1,7	1,3
4	16,7		0,1	11,6	65,7	0,7	9,5
5	27,1		0,1	43,5	38,4	0,2	0,0
6			0,0	2,1	5,9	0,2	1,3
7			0,0	13,2	52,5	0,3	1,3
8			0,2	12,7	48,2	2,7	0,4
9			0,3	26,1	14,8	0,1	0,1
10			0,2	18,6	24,8	1,4	6,5
11			0,0	35,2	9,2	1,4	52,8
12			0,1	22,7	52,0	0,6	0,3
13			0,1	25,7	22,8	0,4	5,5
14			0,0	22,4	10,7	0,7	5,4
15			0,0	12,6		0,5	
16			0,4	5,0		0,0	
17			0,3	7,7		0,0	
18			0,0			5,7	
19			0,0			10,1	
20			0,2			0,1	
21			0,1			1,2	
22			0,0			20,0	
23			0,0			0,9	
24			0,0			16,8	
25			0,0				



Figura 7: Detecção de pessoa no canto esquerdo da imagem.



Figura 8: Detecção de pessoa no canto superior esquerdo da imagem em cena com chuva.

Tendo em vista esses resultados de eficácia da amostra inicial, os fluxos de vídeos que apresentaram desempenho insatisfatório mostram-se inaceitáveis numa pesquisa sobre degradação de desempenho. Assim, fez-se necessário definir um limiar mínimo para a eficácia nos fluxos de vídeos originais possibilitando eliminar os problemas relacionados ao desempenho do algoritmo de detecção utilizado. Com esta decisão, a amostra adotada neste trabalho ficou definida como sendo todos os fluxos de vídeo que tiveram eficácia no detector de objetos superior a 5% e que estivessem na resolução *Full HD*.

Desta forma, a quantidade de fluxos de vídeos da amostra passou de 101 fluxos de vídeo para 48 fluxos de vídeos e a quantidade de cenários foi reduzida de 7 para 6 cenários. Nessa configuração, o cenário 3 foi completamente removido e os cenários 6 e 7 reduzidos a menos da metade dos fluxos de vídeos. Os cenários 4 e 5 foram os únicos cenários que continuaram a possuir mais de 10 elementos e por isso também foram analisados individualmente nas aplicações das hipóteses definidas. Os itens destacados em vermelho na Tabela 1 foram os fluxos de vídeos eliminados da amostra inicial.

Uma análise posterior desta nova amostra revelou uma média de 56,18% dos objetos por fluxo de vídeo com tamanho vertical inferior a 96 *pixels* conforme ilustrado na Figura 9, em que se visualiza a distribuição percentual do tamanho vertical dos objetos por fluxo de vídeo. Dessa forma, considerando que o detector HOG foi previamente treinado com objetos de tamanho vertical superior a 96 *pixels*, a eficácia do algoritmo foi reavaliada desconsiderando os objetos com tamanhos verticais inferiores a 96 *pixels* tanto para a nova amostra quanto nos experimentos realizados posteriormente. Os dados finais da eficácia do algoritmo podem ser observados no Apêndice B.

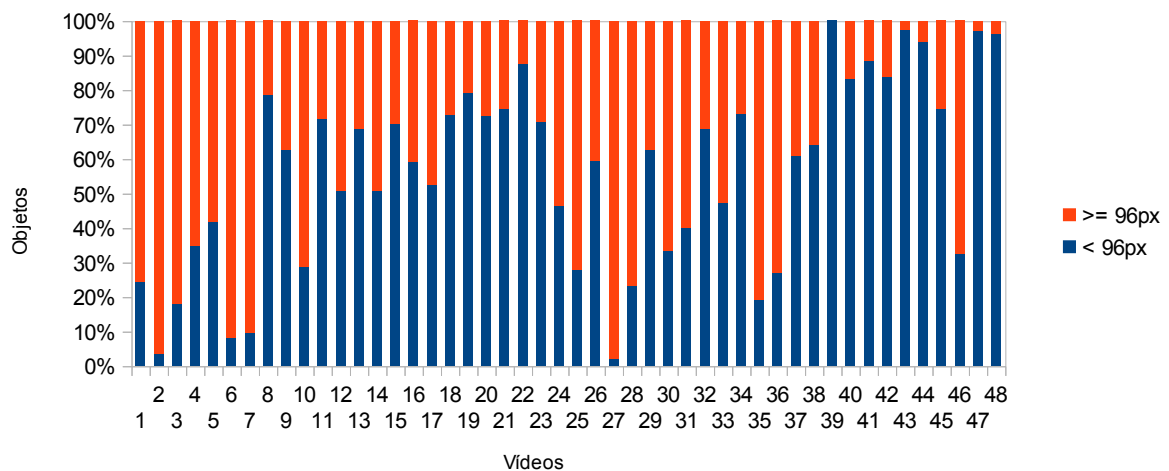


Figura 9: Distribuição dos objetos nos fluxos de vídeos por tamanho vertical em *pixels*.

3.2.3 Tratamentos

Para a definição dos tratamentos a serem utilizados nos experimentos se analisou as taxas de transmissão dos fluxos de vídeos originais e que podem ser observados na Figura 10. Os cenários 4 e 5, por serem os cenários que possuem maiores quantidades de fluxos de vídeos da amostra, foram destacados e nota-se que são cenários com características estatísticas similares, como por exemplo o desvio padrão, e em extremidades opostas de taxa de transmissão em relação à amostra completa.

A partir da análise dos níveis de compactação dos fluxos de vídeos, o valor de 50% da menor taxa de transmissão foi definido como tratamento inicial para a hipótese de compressão do fluxo de vídeo, ou seja, aproximadamente 5 kbps, e então, em seguida, adicionados os tratamentos de 4 kbps, 3 kbps e 2 kbps. Para a hipótese de redução da resolução espacial os tratamentos foram as resoluções HD e *Half Full* HD. A última hipótese começou com os níveis obtidos pelas hipóteses anteriores, aumentando-se o nível de compactação e reduzindo

a resolução espacial dos fluxos de vídeos à medida que os resultados foram sendo obtidos. A Tabela 2 sumariza os tratamentos definidos por hipótese.

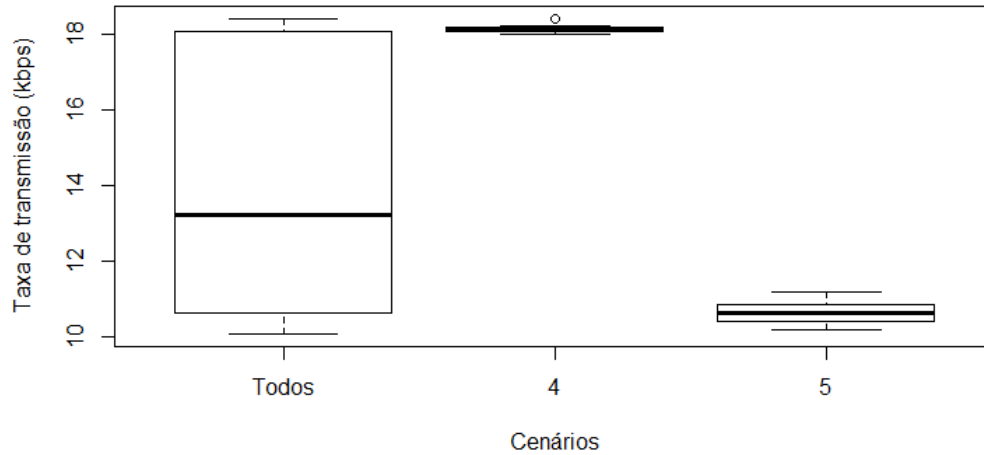


Figura 10: Taxa de transmissão dos fluxos de vídeos da amostra definitiva.

Tabela 2: Tratamentos por hipótese.

Hipótese	Tratamentos
Compressão do fluxo de vídeo	5 kbps, 4 kbps, 3 kbps, 2 kbps
Redução da resolução espacial	HD e Half Full HD
Compressão e redução da resolução espacial	5kbps, 4 kbps, 3kbps em HD

3.2.4 Testes de hipótese

Um teste de hipótese é um método de inferência estatística que utiliza dados provenientes de um estudo científico, ou seja, um procedimento estatístico baseado na análise de uma amostra retirada de uma população sendo utilizados para avaliar determinados parâmetros que são desconhecidos (CASTELLAN JR; SIEGEL, 2006). Os conceitos fundamentais para um teste de hipótese são:

- Hipótese nula: a hipótese que assumimos como verdade para a construção do teste.
- Hipótese alternativa: resultado identificado para o caso da hipótese nula não ter evidência estatística que a defenda.
- Erro tipo I (α): a probabilidade de rejeitarmos a hipótese nula quando ela é efetivamente verdadeira, também chamado de nível de significância.
- Erro tipo II (β): a probabilidade de rejeitarmos a hipótese alternativa quando ela é efetivamente verdadeira.

Neste trabalho foi adotado 5% para o nível de significância (α) em todos os testes de hipóteses.

Os testes de hipóteses são classificados em paramétricos e não paramétricos baseados nas condições que a amostra deve ter para aplicação do teste de hipótese. Uma das principais condições para um teste de hipótese ser classificado como paramétrico é utilizar o pressuposto da distribuição normal dos dados.

Uma das formas de verificar a normalidade dos dados da amostra é utilizar testes de hipóteses que verificam esse pressuposto. A Tabela 3 apresenta os resultados de testes de hipótese que verificam a normalidade dos dados aplicados à amostra desta pesquisa. Os resultados dos testes de hipótese rejeitam a hipótese nula de normalidade dos dados, pois o p-valor é inferior ao nível de significância (α) definido em 5%. Com isso, a avaliação das hipóteses definidas nesta pesquisa deverão utilizar testes de hipótese não paramétricos.

Tabela 3: Teste de normalidade da amostra.

Teste de hipótese	p-valor
Shapiro-Wilk (SHAPIRO; WILK, 1965)	5,271e-05
Anderson-Darling (ANDERSON; DARLING, 1952)	2,697e-05
Lilliefors (Kolmogorov-Smirnov) (LILLIEFORS, 1967)	0,003177

O teste de Friedman (FRIEDMAN, 1937) é um teste não paramétrico utilizado para comprovação da hipótese nula de que k amostras relacionadas provêm da mesma população ou que k tratamentos têm efeitos idênticos. Se o resultado do teste de Friedman é significativo, ou seja, se a hipótese nula é rejeitada, ele indica que pelo menos um dos tratamentos difere de pelo menos outro tratamento. O teste não informa qual dos tratamentos é diferente, nem menciona quantos grupos são diferentes dos demais. Um procedimento adotado para determinar qual tratamento (ou tratamentos) difere é o teste de comparações múltiplas (CASTELLAN JR; SIEGEL, 2006), similar ao teste paramétrico de Tukey (CASTELLAN JR; SIEGEL, 2006) para dados transformados em postos. Ressalta-se que existem comparações múltiplas entre todos os tratamentos (muitos para muitos), e comparações múltiplas de tratamentos com um controle (muitos para um).

Neste trabalho foi adotado o teste de Friedman para identificar as diferenças entre os tratamentos e o teste de comparações múltiplas com um controle para identificar os tratamentos que diferem da amostra original. Ademais, como o interesse é identificar se a eficácia diminui, utilizou-se a opção unicaudal à esquerda para o teste de comparações

múltiplas. As hipóteses nulas formuladas para o teste de comparações múltiplas já considerando os tratamentos definidos podem ser observadas na Tabela 4.

Tabela 4: Hipótese para teste de comparações múltiplas com um controle.

Hipótese de pesquisa	Hipótese para teste
Compressão do fluxo de vídeo	$AUC(\langle Q, c_{original}, r \rangle) \geq AUC(\langle Q, c_i, r \rangle)$ $\forall c_i \in (2, 3, 4, 5) \text{ kbps}$
Redução da resolução espacial	$AUC(\langle Q, c, r_{original} \rangle) \geq AUC(\langle Q, c, r_i \rangle)$ $\forall r_i \in (HD, hFHD)$
Compressão e redução da resolução espacial	$AUC(\langle Q, c_{original}, r_{original} \rangle) \geq AUC(\langle Q, c_k, r_m \rangle)$ $\forall c_k \in (3, 4, 5) \text{ kbps e } r_m = HD$

3.3 CONSIDERAÇÕES FINAIS

As formalizações e resultados preliminares apresentados neste capítulo descrevem a condução da pesquisa realizada neste trabalho. As hipóteses foram representadas matematicamente para em seguida ser descrito a execução experimental e propostos os testes estatísticos apropriados. A execução experimental da amostra inicial indicou deficiências no algoritmo de detecção escolhido que, por consequência, contribuíram para que a amostra inicial fosse filtrada dos fluxos de vídeos que apresentavam condições ambientais adversas.

Os resultados finais obtidos com a execução dos ensaios experimentais e a avaliação desses resultados com as hipóteses e objetivos da pesquisa propostos serão discutidos no próximo capítulo.

4 EXPERIMENTOS E RESULTADOS

Neste capítulo, serão apresentados os resultados obtidos da execução dos experimentos e feita uma análise desses resultados. Na primeira parte, apresentamos os resultados relativos aos ensaios para cada hipótese levantada pela metodologia descrita no capítulo anterior. Na seção seguinte são apresentadas avaliações adicionais sobre esses resultados.

4.1 EXPERIMENTOS

4.1.1 Hipótese da compressão do fluxo de vídeo

O objetivo da hipótese da compressão do fluxo de vídeo é identificar um fator máximo de compressão do fluxo de vídeo tal que, se superado, a eficácia do algoritmo de detecção é reduzida. Ao aumentar a compressão do fluxo de vídeo diminui-se a taxa de transmissão do mesmo permitindo um aumento da escalabilidade de sistemas que utilizam algoritmos de detecção de pessoas.

Os tratamentos definidos para essa hipótese foram as taxas de transmissão de 5 kbps, 4 kbps, 3 kbps e 2 kbps e a taxa de transmissão original dos fluxos de vídeos. A avaliação com o objetivo de identificar a diferença entre quaisquer tratamentos foi feita pelo teste de Friedman, cujos resultados, por cenário, estão sumarizados na Tabela 5. Como o nível de significância (α) foi definido em 5%, os p-valores do cenário completo e do cenário 5 indicam que pelo menos um tratamento difere de outro tratamento.

Tabela 5: Teste de Friedman para a hipótese 1.

Cenários	p-valor
Completo	5,756e-12
4	0,5412
5	1,348e-09

A análise para identificar os tratamentos que diferem da amostra com as taxas de transmissão originais foi realizada pelo teste de múltiplas comparações com um controle, sendo o controle o tratamento com as taxas de transmissão originais. Na coluna de diferença crítica apresentada na Tabela 6 temos o valor máximo de diferença permitido entre os

tratamentos e o controle. Caso a diferença obtida entre um tratamento e o controle seja superior ao da diferença crítica, isto indica que o tratamento difere do controle. Estes casos estão identificados na tabela através da cor vermelha.

Ao se considerar a amostra completa, os resultados evidenciam um valor máximo de compressão dos fluxos de vídeos na taxa de transmissão de 4 kbps. Isso significa que, como a média da taxa de transmissão dos fluxos de vídeos é de 14,069 kbps, foi possível reduzir a taxa de transmissão dos fluxos para aproximadamente 28,5% do seu original. No cenário 4, reforçando o resultado do teste de Friedman, não foi identificada diferença em nenhum dos tratamentos.

Tabela 6: Análise *post-hoc* de múltiplas comparações para a hipótese 1.

Cenários	5 kbps	4 kbps	3 kbps	2 kbps	Dif. crítica
Completo	24	31	53	117	33,46
4	3	16	14	3	19,31
5	17	22	37	54	18,07

Na Figura 11 visualiza-se o comportamento da eficácia do algoritmo de detecção de cada tratamento por cenário. Observa-se que para o cenário completo e o cenário 5, a cada tratamento há uma pequena perda de eficácia do algoritmo de detecção, ao contrário do que ocorre com o cenário 4, reforçando os resultados obtidos nos testes de hipótese apresentados anteriormente.

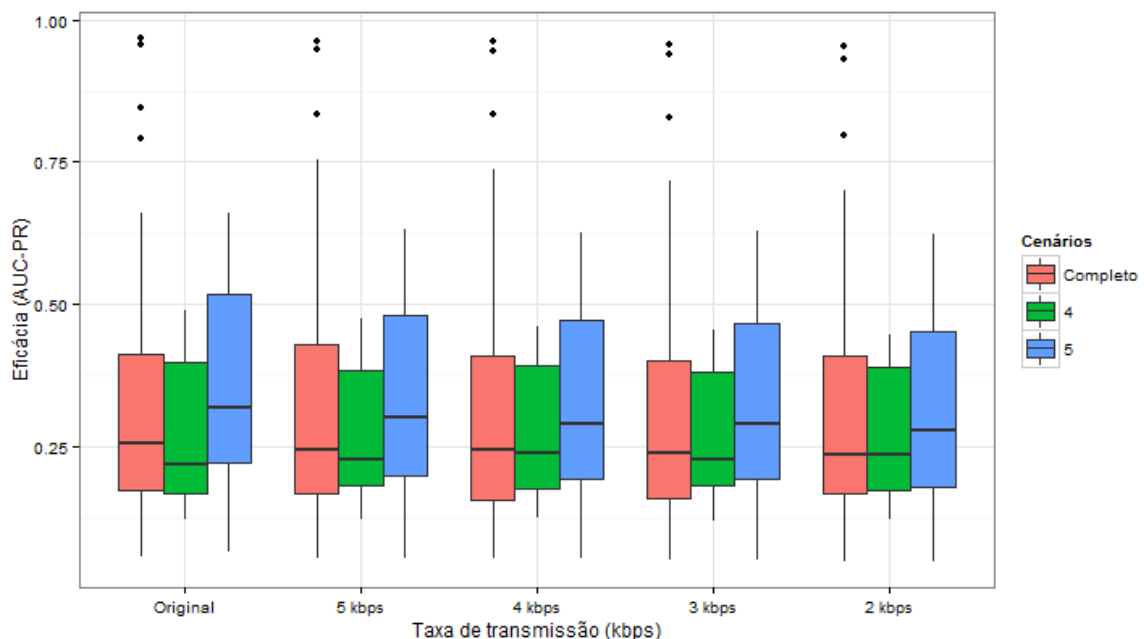


Figura 11: Eficácia da detecção na hipótese 1 sob diferentes níveis de compressão.

4.1.2 Hipótese da redução da resolução espacial

O objetivo da hipótese da redução da resolução espacial do fluxo de vídeo é identificar uma resolução espacial mínima para um fluxo de vídeo tal que, se reduzida, a eficácia do algoritmo de detecção diminui. Ao reduzir a resolução espacial de um fluxo de vídeo diminui-se a quantidade de pixels dos quadros do fluxo de vídeo permitindo que uma mesma taxa de transmissão possa ter um desempenho visual superior do que se a resolução espacial fosse maior.

Os tratamentos definidos para essa hipótese são as resoluções espaciais HD e *Half Full* HD, e a resolução espacial original dos fluxos de vídeos em *Full* HD. A avaliação com o objetivo de identificar a diferença entre quaisquer tratamentos foi feita pelo teste de Friedman, cujos resultados, por cenário, estão sumarizados na Tabela 7. Como o nível de significância (α) foi definido em 5%, os p-valores de todos os cenários indicam que pelo menos um tratamento difere de outro tratamento.

Tabela 7: Teste de Friedman para a hipótese 2.

Cenários	p-valor
Completo	1,724e-06
4	1,724e-06
5	1,724e-06

Tabela 8: Análise *post-hoc* de múltiplas comparações para a hipótese 2.

Cenários	HD	<i>Half Full</i> HD	Dif. crítica
Completo	7	50	18,81
4	9	12	10,86
5	5	10	10,15

A análise para identificar os tratamentos que diferem da amostra com a resolução espacial original foi realizada pelo teste de múltiplas comparações com um controle, onde o controle, sendo o controle o tratamento com a resolução espacial original. Na coluna de diferença crítica apresentada na Tabela 8 temos o valor máximo de diferença permitido entre os tratamentos e o controle. Caso a diferença obtida entre um tratamento e o controle seja

superior ao da diferença crítica, isto indica que o tratamento difere do controle. Estes casos estão identificados na tabela através da cor vermelha.

Ao se considerar a amostra completa, os resultados evidenciam que se encontrou um valor mínimo para a resolução espacial no tratamento com resolução espacial em HD. No cenário 5, embora o resultado do teste de Friedman evidenciou diferença entre os tratamentos, não foi possível identificar nenhum tratamento que diferenciasse do controle, entretanto o tratamento com resolução espacial em *Half Full* HD tem um resultado próximo da diferença crítica obtida com nível de significância (α) de 5% indicando uma limiaridade no resultado deste tratamento.

Na Figura 12 visualiza-se o comportamento da eficácia do algoritmo de detecção de cada tratamento por cenário. Observa-se que, em todos cenários, a cada tratamento há uma pequena perda de eficácia do algoritmo de detecção, com exceção do tratamento *Half Full* HD do cenário 5, em que ocorre também um aumento na variância dos dados.

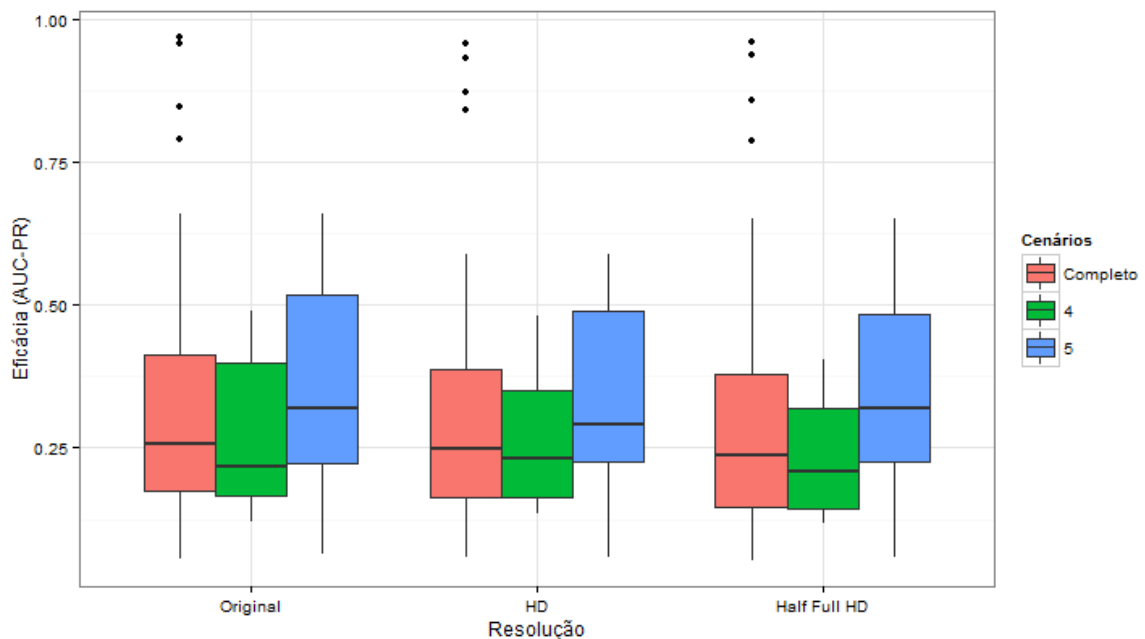


Figura 12: Eficácia da detecção na hipótese 2 sob diferentes resoluções espaciais.

4.1.3 Hipótese conjunta da compressão e redução da resolução espacial

O objetivo da hipótese conjunta da compressão e redução da resolução espacial do fluxo de vídeo é identificar uma combinação de fator máximo de compressão e de resolução espacial mínima tal que, se o fator de compressão for aumentado ou a resolução espacial for reduzida, a eficácia do algoritmo de detecção diminui. Busca-se ao reduzir a resolução

espacial de um fluxo de vídeo diminuir a quantidade de pixels dos quadros do fluxo de vídeo permitindo então um fator de compressão maior sem que isso comprometa a eficácia do algoritmo de detecção.

Os tratamentos definidos para essa hipótese são as taxas de transmissão de 5 kbps, 4 kbps e 3 kbps, todos com a resolução espacial HD, e do tratamento com a taxa de transmissão e a resolução espacial originais dos fluxos de vídeos. A avaliação com o objetivo de identificar a diferença entre quaisquer tratamentos foi feita pelo teste de Friedman, cujos resultados, por cenário, estão sumarizados na Tabela 9. Como o nível de significância (α) foi definido em 5%, os p-valores do cenário completo e do cenário 5 indicam que pelo menos um tratamento difere de outro tratamento.

Tabela 9: Teste de Friedman para a hipótese 3.

Cenários	p-valor
Completo	1,233e-07
4	0.5672
5	0,0001586

A análise para identificar os tratamentos que diferem da amostra com as taxas de transmissão e resolução espacial original foi realizada pelo teste de múltiplas comparações com um controle, sendo o controle o tratamento com as taxas de transmissão e resolução espacial original. Na coluna de diferença crítica apresentada na Tabela 10 temos o valor máximo de diferença permitido entre os tratamentos e o controle. Caso a diferença obtida entre um tratamento e o controle seja superior ao da diferença crítica, isto indica que o tratamento difere do controle. Estes casos estão identificados na tabela através da cor vermelha.

Tabela 10: Análise *post-hoc* de múltiplas comparações para a hipótese 3.

Cenários	5 kbps @ HD	4 kbps @ HD	3 kbps @ HD	Dif. crítica
Completo	3	5	68	26,05
4	10	3	9	15,04
5	7	8	29	14,07

Ao considerarmos a amostra completa, os resultados indicam que se encontrou um valor máximo de compressão dos fluxos de vídeos em 4 kbps na resolução *Full HD*. Isso

significa que não foi possível superar o aumento do fator de compressão, quando foi aplicado isoladamente, ao se reduzir conjuntamente a resolução espacial. No cenário 4, reforçando o resultado do teste de Friedman, não foi identificada diferença em nenhum dos tratamentos.

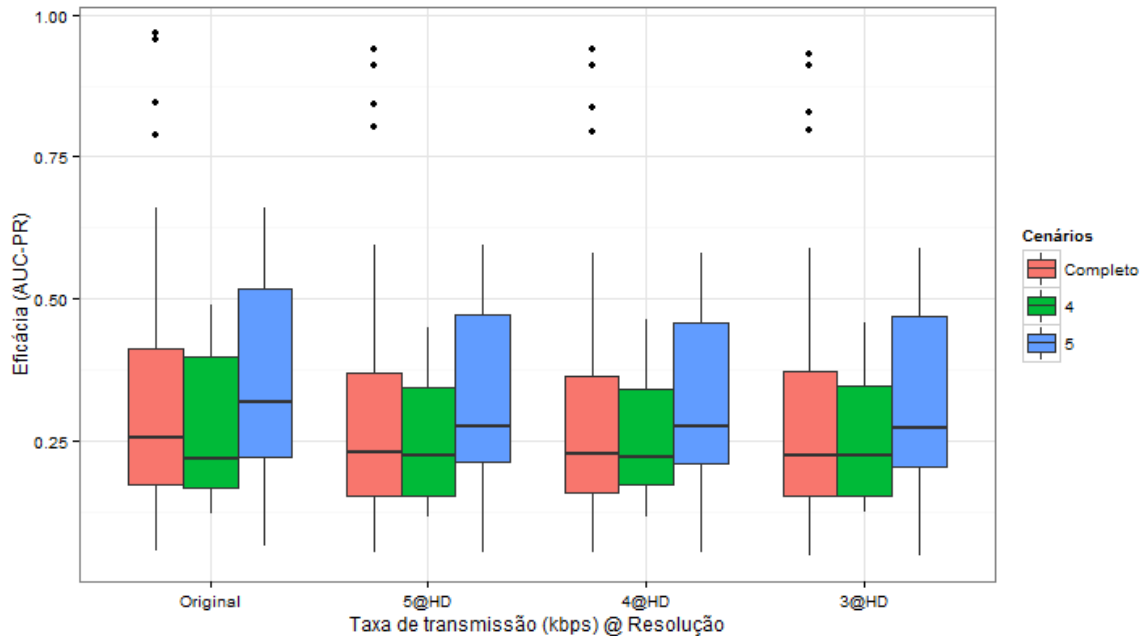


Figura 13: Eficácia da detecção na hipótese 3 sob diferentes níveis de compressão e resolução.

Na Figura 13 visualiza-se o comportamento da eficácia do algoritmo de detecção de cada tratamento por cenário. Observa-se que para os cenários completo e para o cenário 5, a cada tratamento há uma pequena perda de eficácia do algoritmo de detecção, ao contrário do que ocorre com o cenário 4, reforçando os resultados obtidos nos testes de hipótese apresentados anteriormente.

4.2 AVALIAÇÕES ADICIONAIS

Uma análise adicional que compara os resultados das hipóteses 1 e 3 produziu uma relação entre o nível de compactação e a redução da eficácia de detecção que pode ser observada na Figura 14. Para os fluxos de vídeos em *Full HD*, embora a análise estatística aponte diferença entre os tratamentos, conforme resultados apresentados na Tabela 6, percebe-se uma perda média inferior a 1% mesmo com uma redução de 50% da taxa de transmissão do fluxo de vídeo entre os resultados. Entretanto quando ocorre a redução da resolução espacial do vídeo para HD, percebe-se logo uma queda na eficácia.

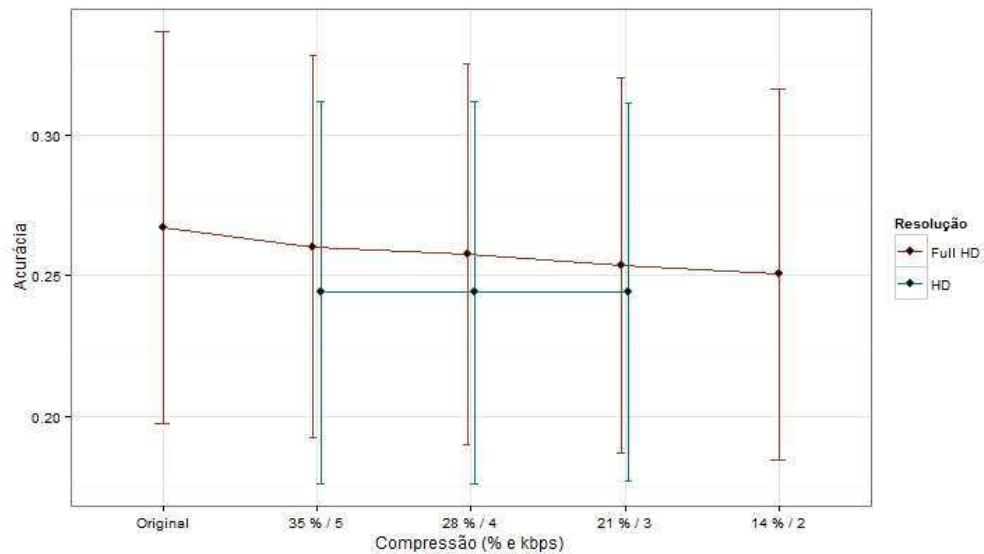


Figura 14: Eficácia da detecção sob diferentes níveis de compressão e resoluções espaciais.

O comparativo visual entre os tratamentos com taxa de transmissão de 4 kbps indicam uma superioridade em eficácia quanto maior for a resolução espacial. O teste de Friedman aplicado apenas aos 2 tratamentos tem como resultado um p-valor de 0,003892, que por ser inferior ao nível de significância (α) definido em 5%, confirma a superioridade em eficácia do tratamento que não altera a resolução espacial original dos fluxos de vídeos.

Um dos possíveis fatores que provocam esse resultado pode ser observado nas Figuras 15 e 16, onde a primeira é a imagem original e a segunda é a imagem após o redimensionamento espacial, ambas com um objeto de interesse em destaque. Observa-se que o objeto quando em resolução maior possui mais *pixels* para visualização, permitindo uma melhor identificação do objeto. No entanto, quando o fluxo de vídeo tem sua resolução espacial reduzida, a quantidade de pixels desse mesmo objeto diminui, provocando prejuízo na identificação do objeto, e por isso, refletindo na eficácia do algoritmo de detecção.



Figura 15: Exemplo de imagem no formato original com objeto destacado



Figura 16: Exemplo de imagem redimensionada com objeto destacado

Outra possível fator que causaria este resultado vem exatamente quando analisamos comparativamente os resultados do cenários 4 e cenário 5. O cenário 4 teve uma resposta experimental oposta aos resultados da amostra completa, enquanto que o cenário 5, teve uma resposta experimental similar à amostra completa.

Ao analisar os fluxos de vídeos do cenário 4 e do cenário 5 é possível observar que a tomada de câmera do cenário 4 apresenta os objetos mais próximos, com uma horizontalidade aproximada das ações dos objetos e a câmera não sofre interferências externas como vibrações. No entanto, o cenário 5, apresenta uma maior quantidade de objetos ao fundo da imagem pois a câmera está distante das ações dos objetos, numa tomada superior, e por fim, em vários momentos ocorrem vibrações na câmera devido a fatores externos causadas por ventilação natural. As Figuras 17 e 18 ilustram as diferenças nas tomadas das câmeras e as proximidades dos objetos.



Figura 17: Exemplo de imagem do cenário 4.



Figura 18: Exemplo de imagem do cenário 5.

Faz-se necessário destacar também que os fluxos de vídeos do cenário 4 foram mais impactados proporcionalmente com a compressão do fluxo de vídeo que os fluxos de vídeos do cenário 5 porque tinham originalmente uma média de compactação inferior como pode ser observado na comparação da Figura 10.

Por fim, o comportamento padrão esperado para quando ocorre aumento do nível de compactação do fluxo de vídeo é a redução da eficácia do algoritmo de detecção, entretanto os resultados do cenário 4 foram contrários a esse comportamento. Um fator que pode ter causado esse resultado foi a suavização de detalhes da imagem, *e.g.* objetos imóveis ou de fundo, pelo codificador, ao se aumentar o nível de compactação dos fluxos de vídeo. No entanto, com aumentos sucessivos do nível de compactação, o codificador será obrigado a utilizar os demais elementos da imagem na compactação. Para confirmar esta hipótese, foram estendidos os experimentos no cenário 4, com objetivo de obter a curva de degradação da eficácia conforme se aumentava o nível de compressão.

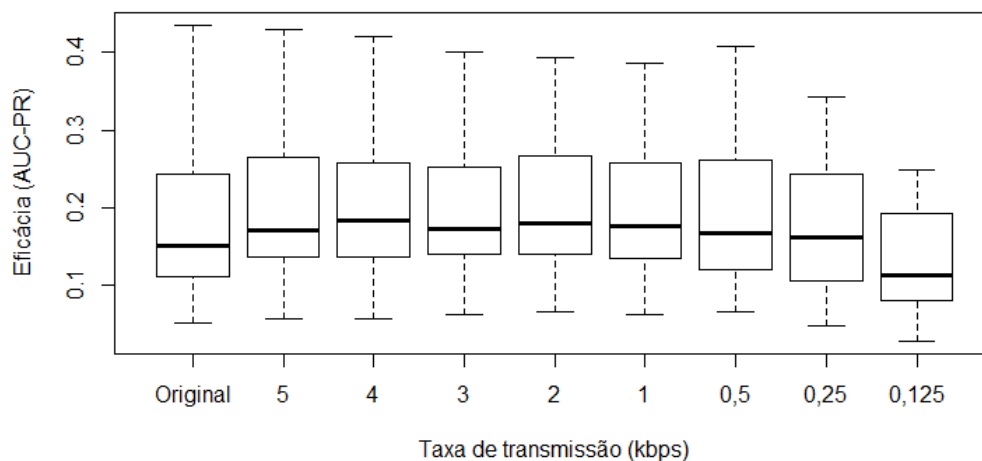


Figura 19: Eficácia da detecção no cenário 4 sob diferentes níveis de compressão

A Figura 19 ilustra a curva de degradação da eficácia a cada nível de compactação. Observa-se que, a partir da taxa de transmissão de 1 kbps, a eficácia do detector começa a apresentar degradação à medida em que se aumenta o nível de compactação. Assim sendo, possivelmente existe um nível crítico de compactação próximo a taxa de transmissão de 1 kbps para os fluxos de vídeos do cenário 4.

4.3 CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentados os resultados finais obtidos com a execução dos ensaios experimentais seguindo a metodologia proposta. A avaliação desses resultados e seus desdobramentos indicaram que objetivos propostos na pesquisa foram alcançados.

No próximo capítulo as conclusões e limitações desta pesquisa serão detalhadas juntamente com uma apresentação de trabalhos futuros que poderão ser desenvolvidos de forma a aprimorar os resultados apresentados.

5 CONCLUSÃO

Esta pesquisa avaliou o impacto da redução da taxa de transmissão dos fluxos de vídeos, utilizados em sistemas inteligentes de videovigilância, na eficácia da detecção de pessoas dos algoritmos empregados nesses sistemas. A principal contribuição deste trabalho foi identificar um fator de compressão máximo para vídeos na resolução *Full HD* em que o algoritmo de detecção de pessoas do tipo HOG (DALAL; TRIGGS, 2005) não teve sua eficácia comprometida.

Nesse sentido, foi possível reduzir a taxa de transmissão do fluxo de vídeo, sem perdas na eficácia do detector, na média, para 28% da taxa de transmissão original, o que para sistemas de videovigilância distribuídos permitiria a alocação de mais câmeras no sistema ou o compartilhamento da infraestrutura de comunicação com outros serviços. Um segundo resultado indica que uma redução para 14% da taxa de transmissão degrada, na média, menos de 2% na eficácia do detector.

Uma contribuição adicional dessa dissertação foi com a utilização da base de vídeos VIRAT na versão 2.0 (OH et al., 2011). Essa base de vídeos mostrou-se adequada para a realização de experimentos envolvendo a detecção de pessoas para sistemas de videovigilância que atualmente utilizam câmeras de alta resolução. O algoritmo HOG encontrou em vários vídeos, dificuldades na detecção, ficando com uma média de 26% na métrica AUC-PR, valor inferior ao obtido em 2005 (EVERINGHAM; ZISSERMAN, 2006) de 43% no Pascal VOC *Challenge*.

Por fim, com base no método experimental, não foi possível combinar a compactação do fluxo de vídeo com a redução da resolução espacial. Possivelmente, esse resultado deve-se a objetos que, quando em resolução maior, possuem mais pixels para visualização, então, quando o fluxo de vídeo tem sua resolução espacial reduzida, a quantidade de pixels desse mesmo objeto não permite mais ao algoritmo de detecção identificar o objeto. Um dos fatores que permitiram essa confirmação foi a base de vídeos escolhidos que continha muitos objetos identificados no fundo das imagens com poucos *pixels* de tamanho.

5.1 LIMITAÇÕES

Na interpretação dos resultados faz-se necessário apontar algumas limitações da abrangência da aplicação destes resultados. O primeiro aspecto a se considerar vem exatamente da utilização do detector HOG do OpenCV. Os conjuntos de imagens utilizados

no treinamento do detector eram de imagens com resoluções inferiores HD exatamente como descrito em DALAL (2005). Não seria totalmente errôneo considerar que, caso se utilizasse um conjunto de imagens com grau de dificuldade similar àquele da base de vídeos utilizada nesta pesquisa, *i.e.*, imagens em alta definição com objetos pequenos, o detector obtivesse resultados superiores aos reportados.

Essa limitação no detector derivou um problema que impediu que se utilizasse um conjunto amostral maior de vídeo da base VIRAT (OH et al., 2011), o que poderia favorecer em responder questionamentos numa comparação entre cenários, como feito na análise dos resultados no capítulo anterior. Entretanto, a utilização de uma amostra maior também poderia apontar para uma piora nos resultados obtidos.

Um segundo aspecto a se considerar vem da utilização de apenas uma base de vídeos. Esta limitação pode indicar a utilização de fluxos de vídeos provenientes de um mesmo conjunto de *hardware* de gravação, sendo que, não é incomum notar diferenças em imagens provenientes de diferentes *hardwares* devido a diferenças técnicas nos sensores de obtenção das imagens das câmeras. Essas diferenças nas imagens poderiam prejudicar a eficácia do detector e conseqüentemente os resultados finais.

5.2 TRABALHOS FUTUROS

Diante dos resultados apresentados neste trabalho surgem, naturalmente, novas ideias para melhoramento dos experimentos, desenvolvimentos de novas hipóteses e linhas de pesquisa. Neste sentido, estão listados a seguir alguns dos trabalhos futuros sugeridos para a continuidade da pesquisa em sistemas de vigilância automática:

a) Ampliar a diversidade das resoluções espaciais iniciais dos fluxos de vídeos para HD, e 4K e 8K UHD;

Assim como o custo das câmeras *Full* HD vem caindo consideravelmente permitindo a sua utilização nos sistemas de vigilância atuais é provável que os sistemas de vigilância futuros façam uso de câmeras UHD conforme ocorra o barateamento desses dispositivos. O objetivo dessa linha de pesquisa seria ampliar e confirmar as contribuições deste trabalho quando os fluxos de vídeos originais estiverem em resoluções diferentes da *Full* HD.

b) Acrescentar a avaliação da eficácia de um algoritmo estado-da-arte em detecção de pessoas baseado na estratégia de reconhecimento por modelos parciais de objetos (FELZENSZWALB et al., 2010);

Esta pesquisa utilizou um dos algoritmos base em detecção de objetos, no entanto, vários outros algoritmos e técnicas existem, inclusive alguns sendo considerados aperfeiçoamentos do algoritmo HOG, como é o caso da detecção por modelos parciais de objetos (FELZENSZWALB et al., 2010). A reprodução desta pesquisa, utilizando estes algoritmos derivados do HOG, poderia confirmar os resultados obtidos e generalizá-los para todo algoritmo que utilize HOG como parte do método de detecção de objetos.

c) Treinar o algoritmo por meio da base de vídeos utilizadas nos experimentos e realizar uma validação cruzada para obter os resultados;

A utilização do algoritmo HOG com o treinamento disponível no OpenCV, que é baseado nos trabalhos de DALAL (2005), limitou a utilização da base de vídeos VIRAT devido ao desempenho obtido em vários fluxos de vídeos. Uma forma de aperfeiçoar os resultados, e com isso utilizar os demais fluxos de vídeos da base VIRAT, seria realizar o treinamento e a verificação dos experimentos através de uma validação cruzada.

d) Utilizar uma segunda base de vídeos proporcionando uma maior independência na qualidade das câmeras utilizadas.

A limitação de se utilizar fluxos de vídeos provenientes de um mesmo conjunto de hardware de gravação pode ser mitigada com esse ajuste metodológico. A utilização de bases de vídeos produzidas por diferentes autores melhoraria então a validade dos resultados encontrados.

REFERÊNCIAS

ADAMS, A. A.; FERRYMAN, J. M. The future of video analytics for surveillance and its ethical implications. **Security Journal**, 14 jan. 2013.

AHMEDALI, T.; CLARK, J. J. **Collaborative Multi-Camera Surveillance with Automated Person Detection**The 3rd Canadian Conference on Computer and Robot Vision (CRV'06). **Anais...IEEE**, 2006.

AMAZON. **Amazon Elastic MapReduce**, 2009. Disponível em: <<http://aws.amazon.com/elasticmapreduce/>>. Acesso em: 17 nov. 2014.

ANDERSON, T. W.; DARLING, D. A. Asymptotic Theory of Certain “Goodness of Fit” Criteria Based on Stochastic Processes. *The Annals of Mathematical Statistics*, v. 23, n. 2, p. 193–212, jun. 1952.

ANSI. **C Language**, X3.1591430 Broadway, New York, NY 10018, USA, 1989.

BRADSKI, G. The OpenCV Library. **Dr. Dobb’s Journal of Software Tools**, 2000.

CALAVIA, L. et al. A semantic autonomous video surveillance system for dense camera networks in Smart Cities. **Sensors (Basel, Switzerland)**, v. 12, n. 8, p. 10407–29, jan. 2012.

CASTELLAN JR, J.; SIEGEL, S. *Estatística não paramétrica para ciências do comportamento*. Porto Alegre: Artmed, 2006.

CHATTOPADHYAYR, D. et al. **Event Driven Video Surveillance System using City Cloud A solution compliant with Sensor Web Enablement Architecture**Proceedings of the First International Conference on Intelligent Infrastructure at the 47th Annual National Convention COMPUTER SOCIETY of INDIA CSI - 2012. **Anais...2013**.

CHEN, S. et al. A Hierarchical Model Incorporating Segmented Regions and Pixel Descriptors for Video Background Subtraction. **IEEE Transactions on Industrial Informatics**, v. 8, n. 1, p. 118–127, fev. 2012.

DADASHI, N. **Automatic Surveillance and CCTV Operator Workload**. [s.l.] University of Nottingham, 2008.

DALAL, N.; TRIGGS, B. **Histograms of Oriented Gradients for Human Detection**2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). **Anais...IEEE**, 2005.

DIRNITRIJEVIC, Z.; CHANG, E. Y. **SFINX: a multi-sensor fusion and mining system**Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint. **Anais...IEEE**, 2003.

ENZWEILER, M.; GAVRILA, D. M. Monocular pedestrian detection: survey and experiments. **IEEE transactions on pattern analysis and machine intelligence**, v. 31, n. 12, p. 2179–95, dez. 2009.

ESKICIOGLU, A. M.; FISHER, P. S. Image quality measures and their performance. **IEEE Transactions on Communications**, v. 43, n. 12, p. 2959–2965, 1995.

EVERINGHAM, M. et al. The Pascal Visual Object Classes (VOC) Challenge. **International Journal of Computer Vision**, v. 88, n. 2, p. 303–338, 9 set. 2009.

EVERINGHAM, M.; ZISSERMAN, A. The 2005 pascal visual object classes challenge. **Selected Proceedings of the First PASCAL Challenges Workshop**, 2006.

FELZENSZWALB, P. F. et al. Object detection with discriminatively trained part-based models. **IEEE transactions on pattern analysis and machine intelligence**, v. 32, n. 9, p. 1627–45, set. 2010.

FOUNDATION, T. E. **Eclipse**, 2014. Disponível em: <<http://www.eclipse.org/org/>>. Acesso em: 17 nov. 2014.

FRIEDMAN, M. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *Journal of the American Statistical Association*, v. 32, n. 200, p. 675, dez. 1937.

GIRGENSOHN, A. et al. **DOTS: support for effective video surveillance** Proceedings of the 15th international conference on Multimedia - MULTIMEDIA '07. **Anais...**New York, New York, USA: ACM Press, 2007.

GONZALEZ, R. C.; WOODS, R. E. **Processamento de imagens digitais**. [s.l.] Edgard Blucher, 2000.

GUBBI, J. et al. Internet of Things (IoT): A vision, architectural elements, and future directions. **Future Generation Computer Systems**, v. 29, n. 7, p. 1645–1660, set. 2013.

HARRIS, C.; STEPHENS, M. A combined corner and edge detector. **Alvey vision conference**, v. 15, p. 147–152, 1988.

HERNÁNDEZ-MUÑOZ, J.; VERCHER, J.; MUÑOZ, L. Smart cities at the forefront of the future internet. **The future internet**, p. 447–462, 2011.

HU, W. et al. A Survey on Visual Surveillance of Object Motion and Behaviors. **IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)**, v. 34, n. 3, p. 334–352, ago. 2004.

JAVED, O. et al. **KNIGHT: a real time surveillance system for multiple and non-overlapping cameras**. 2003 International Conference on Multimedia and Expo. ICME '03. Proceedings (Cat. No.03TH8698). **Anais...**IEEE, 2003.

JI, X. et al. Advances in View-Invariant Human Motion Analysis: A Review. **IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)**, v. 40, n. 1, p. 13–24, jan. 2010.

JOSHI, S. S.; PANCHAL, J. R. Background Subtraction Based Detection and Tracking Of People In Video. v. 1, n. 5, p. 237–244, 2014.

KAFETZAKIS, E. et al. **The Impact of Video Transcoding Parameters on Event Detection for Surveillance Systems** 2013 IEEE International Symposium on Multimedia. **Anais...IEEE**, dez. 2013 Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6746816>>. Acesso em: 14 nov. 2014.

KORSHUNOV, P.; OOI, W. T. **Critical video quality for distributed automated video surveillance** Proceedings of the 13th annual ACM international conference on Multimedia - MULTIMEDIA '05. **Anais...New York, New York, USA: ACM Press, 2005.**

KORSHUNOV, P.; OOI, W. T. Video quality for face detection, recognition, and tracking. **ACM Transactions on Multimedia Computing, Communications, and Applications**, v. 7, n. 3, p. 1–21, 1 ago. 2011.

LANZA, A.; SALTI, S.; STEFANO, L. DI. Background subtraction by non-parametric probabilistic clustering. **Advanced Video and Signal- ...**, n. 1, p. 243–248, 2011.

LILLIEFORS, H. W. On the Kolmogorov-Smirnov Test for Normality with Mean and Variance Unknown. *Journal of the American Statistical Association*, v. 62, n. 318, p. 399–402, jun. 1967.

MARKETRESEARCH.COM. **Global Video Surveillance Market, Applications and Management Services Forecasts (2010-2015)**. Disponível em: <<http://www.marketresearch.com/MarketsandMarkets-v3719/Global-Video-Surveillance-Applications-Management-6083117/>>. Acesso em: 17 nov. 2014.

NAIR, V.; CLARK, J. J. Automated visual surveillance using hidden markov models. **International Conference on Vision Interface**, v. 93, 2002.

OH, S. et al. **A large-scale benchmark dataset for event recognition in surveillance video** CVPR 2011. **Anais...IEEE**, jun. 2011.

PAUL, M.; HAQUE, S.; CHAKRABORTY, S. Human detection in surveillance videos and its applications-a review. **EURASIP Journal on Advances in ...**, p. 1–16, 2013.

PAVLIDIS, I. et al. Urban surveillance systems: from the laboratory to the commercial world. **Proceedings of the IEEE**, v. 89, n. 10, p. 1478–1497, 2001.

POPOOLA, O. P.; WANG, K. Video-Based Abnormal Human Behavior Recognition---A Review. **IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)**, v. PP, n. 99, p. 1–14, 2012.

R CORE TEAM. **R: A Language and Environment for Statistical Computing** Vienna, Austria, 2014. Disponível em: <<http://www.r-project.org>>. Acesso em: 17 nov. 2014.

RÁTY, T. D. Survey on Contemporary Remote Surveillance Systems for Public Safety. **IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)**, v. 40, n. 5, p. 493–515, set. 2010.

REEVE, T. **BSIA attempts to clarify question of how many CCTV cameras there are in the UK**. Disponível em: <<http://www.securitynewsdesk.com/2013/07/11/bsia-attempts-to-clarify-question-of-how-many-cctv-cameras-in-the-uk/>>. Acesso em: 17 nov. 2014.

ROSSUM, G. VAN; DRAKE, F. L. **Python Reference Manual** Virginia, USA, 2001. Disponível em: <<http://www.python.org>>. Acesso em: 18 nov. 2014.

ROWLEY, H.; BALUJA, S.; KANADE, T. Neural network-based face detection. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 20, n. 1, p. 23–38, 1998.

SHAPIRO, S. S.; WILK, M. B. An analysis of variance test for normality (complete samples). *Biometrika*, v. 52, n. 3-4, p. 591–611, 1 dez. 1965.

SHEKIN, D. J. **Handbook of parametric and nonparametric statistical procedures**. [s.l.] crc Press, 2003.

SMEATON, A. F.; OVER, P.; KRAAIJ, W. **Evaluation campaigns and TRECVID** Proceedings of the 8th ACM international workshop on Multimedia information retrieval - MIR '06. **Anais...** New York, New York, USA: ACM Press, 2006 Disponível em: <<http://portal.acm.org/citation.cfm?doid=1178677.1178722>>. Acesso em: 18 nov. 2014.

SMPTE (SOCIETY OF MOTION PICTURE AND TELEVISION ENGINEERS). **SMPTE 296M: Standard for Television—1280 x 720 Progressive Image Sample Structure, Analog and Digital Representation and Analog Interface**. New York, New York, USA. 2001.

SMPTE (SOCIETY OF MOTION PICTURE AND TELEVISION ENGINEERS). **SMPTE 274M: Standard for Television — 1920 x 1080 Scanning and Analog and Parallel Digital Interfaces for Multiple Picture Rates**. New York, New York, USA. 2003.

SONG, M.; TAO, D.; MAYBANK, S. Sparse Camera Network for Visual Surveillance--A Comprehensive Survey. **arXiv preprint arXiv:1302.0446**, p. 1–42, 2013.

TEAM, FF. **FFmpeg**, 2000a. Disponível em: <<http://ffmpeg.org>>. Acesso em: 17 nov. 2014.

TEAM, IA. Image Arcade, 2014. Disponível em: <<http://imgarcade.com/>>. Acesso em: 18 nov. 2014.

TEAM, MP. **MPlayer**, 2000b. Disponível em: <<http://mplayerhq.hu>>. Acesso em: 18 nov. 2014.

TEAM, X. **Xine project**, 2000c. Disponível em: <<http://xine-project.org>>.

- TIAN, Y. et al. Robust Detection of Abandoned and Removed Objects in Complex Surveillance Videos. **IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)**, v. 41, n. 5, p. 565–576, set. 2011.
- TOMASI, C.; KANADE, T. **Detection and tracking of point features**. [s.l.] School of Computer Science, Carnegie Mellon Univ. Pittsburgh, 1991.
- VALERA, M.; VELASTIN, S. Intelligent distributed surveillance systems: a review. **IEE Proceedings - Vision, Image, and Signal Processing**, v. 152, n. 2, p. 192, 2005.
- VIDEOLAN. **VLC media player**, 2001. Disponível em: <<http://www.videolan.org/vlc/>>. Acesso em: 17 nov. 2014.
- VIOLA, P.; JONES, M. J. Robust Real-Time Face Detection. **International Journal of Computer Vision**, v. 57, n. 2, p. 137–154, maio 2004.
- WANG, X. Intelligent multi-camera video surveillance: A review. **Pattern Recognition Letters**, v. 34, n. 1, p. 3–19, jan. 2013.
- WANG, Z. et al. Image Quality Assessment: From Error Visibility to Structural Similarity. **IEEE Transactions on Image Processing**, v. 13, n. 4, p. 600–612, abr. 2004.
- YUAN, X. et al. **A distributed visual surveillance system** Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance, 2003. **Anais...IEEE Comput. Soc**, 2003 Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1217922>. Acesso em: 17 nov. 2014.
- ZHU, M. Recall, precision and average precision. **Department of Statistics and Actuarial Science, University of Waterloo, Waterloo**, 2004.

APÊNDICE A: MAPEAMENTO DE ARQUIVOS DE VÍDEO E NUMERAÇÕES

Neste apêndice os arquivos de vídeos da base VIRAT utilizados nesta pesquisa foram mapeados para uma numeração de forma a facilitar a tabulação dos resultados obtidos conforme pode ser observado na Tabela 11 e no Apêndice B.

Tabela 11: Mapeamento dos arquivos da base VIRAT com numeração utilizada

Numeração	Vídeo	Numeração	Vídeo
1	000001	25	050000_03_000585_000639
2	000002	26	050000_04_000640_000690
3	000003	27	050000_05_000696_000732
4	000004	28	050000_06_000908_000970
5	000006	29	050000_07_001014_001126
6	000101	30	050000_08_001235_001295
7	000102	31	050000_09_001310_001373
8	040103_00_000000_000120	32	050000_10_001462_001491
9	040103_01_000132_000195	33	050000_11_001530_001576
10	040103_02_000199_000279	34	050000_12_001591_001619
11	040103_03_000284_000425	35	050000_13_001722_001766
12	040103_05_000729_000804	36	050000_16_001947_001989
13	040103_07_001011_001093	37	050201_00_000012_000116
14	040103_08_001475_001512	38	050202_10_002159_002233
15	040104_00_000120_000224	39	050203_00_000023_000097
16	040104_01_000227_000457	40	050203_07_001288_001531
17	040104_02_000459_000721	41	050203_09_001960_002083
18	040104_04_000854_000934	42	050300_01_000148_000396
19	040104_05_000939_001116	43	050300_04_001057_001122
20	040104_06_001121_001241	44	050300_10_002176_002238
21	040104_07_001268_001348	45	050301_00_000000_000036
22	040104_08_001353_001470	46	050301_02_000544_000607
23	040104_09_001475_001583	47	050301_03_000933_001046
24	050000_01_000207_000361	48	050000_03_000585_000639

APÊNDICE B: RESULTADO DOS ENSAIOS DAS HIPÓTESES

Neste apêndice estão os dados da eficácia do algoritmo de detecção obtidos para cada cenário e níveis. Para facilitar a reprodução dos resultados estão separados pelas hipóteses em que foram utilizados para os testes estatísticos.

RESULTADOS DO ENSAIO DA HIPÓTESE 1

Tabela 12: Eficácia da detecção do ensaio da hipótese 1

Vídeos	Cenários	Tratamentos					
		Original	Original Filtrado	5 kbps	4 kbps	3 kbps	2 kbps
1	1	75,27%	78,96%	75,48%	73,73%	71,74%	69,85%
2	1	84,18%	84,59%	83,28%	83,28%	82,71%	79,72%
3	1	5,62%	5,68%	5,75%	5,70%	5,54%	5,66%
4	1	16,72%	17,32%	17,09%	15,71%	15,34%	14,43%
5	1	27,08%	29,28%	26,88%	26,49%	25,73%	25,34%
6	2	95,08%	95,64%	94,83%	94,53%	93,84%	93,17%
7	2	96,04%	96,81%	96,30%	96,22%	95,76%	95,43%
8	4	6,28%	12,62%	14,56%	14,95%	15,59%	15,02%
9	4	16,94%	20,22%	21,07%	23,60%	21,47%	22,81%
10	4	10,35%	12,18%	15,40%	15,31%	15,91%	17,14%
11	4	11,65%	17,75%	20,43%	20,34%	21,19%	20,53%
12	4	43,50%	48,94%	47,43%	46,13%	43,87%	42,86%
13	4	13,24%	19,04%	19,02%	18,49%	18,87%	17,22%
14	4	12,74%	17,43%	20,08%	19,97%	19,48%	19,06%
15	4	26,13%	33,81%	35,22%	34,57%	34,07%	32,97%
16	4	18,56%	23,80%	25,60%	25,40%	25,07%	25,33%
17	4	35,20%	42,39%	44,22%	45,89%	45,37%	44,71%
18	4	22,74%	40,15%	42,51%	39,86%	38,57%	40,35%
19	4	25,71%	40,89%	39,91%	39,40%	38,68%	39,73%

Vídeos	Cenários	Tratamentos					
		Original	Original Filtrado	5 kbps	4 kbps	3 kbps	2 kbps
20	4	22,39%	39,68%	37,92%	39,02%	37,75%	38,54%
21	4	12,58%	23,39%	24,36%	24,20%	23,91%	24,02%
22	4	5,04%	14,63%	14,58%	14,29%	14,85%	15,35%
23	4	7,67%	13,17%	12,10%	12,30%	11,94%	12,18%
24	5	28,17%	32,63%	31,21%	30,62%	30,42%	30,12%
25	5	42,88%	47,10%	45,51%	43,95%	44,23%	42,89%
26	5	25,05%	31,10%	29,10%	27,49%	27,68%	25,26%
27	5	65,71%	66,03%	63,10%	62,50%	62,98%	62,42%
28	5	38,39%	40,53%	35,91%	36,33%	33,71%	34,03%
29	5	5,85%	6,39%	5,41%	5,19%	5,03%	4,72%
30	5	52,52%	58,87%	55,50%	53,82%	52,82%	52,74%
31	5	48,20%	55,00%	48,74%	48,29%	47,55%	45,82%
32	5	14,85%	21,52%	19,01%	18,67%	18,64%	17,39%
33	5	24,76%	30,08%	26,54%	26,31%	25,05%	24,69%
34	5	9,17%	12,89%	12,13%	11,66%	12,01%	11,44%
35	5	51,95%	53,31%	50,64%	50,77%	49,63%	50,77%
36	5	22,76%	24,34%	21,77%	20,86%	20,63%	19,66%
37	5	10,69%	15,67%	14,70%	15,00%	14,22%	13,83%
38	6	10,46%	12,63%	11,97%	11,73%	11,20%	11,61%
39	6	5,72%	11,54%	9,74%	9,82%	8,76%	9,98%
40	6	10,07%	14,25%	10,01%	9,48%	9,88%	10,76%
41	6	20,00%	26,81%	24,66%	24,47%	23,92%	22,66%
42	6	16,78%	24,37%	23,11%	22,95%	21,84%	20,82%
43	7	8,13%	29,20%	23,28%	23,93%	22,50%	23,12%
44	7	9,45%	29,09%	26,95%	26,47%	26,00%	24,08%
45	7	6,55%	19,84%	15,35%	12,54%	13,63%	6,92%

Vídeos	Cenários	Tratamentos					
		Original	Original Filtrado	5 kbps	4 kbps	3 kbps	2 kbps
46	7	52,82%	54,20%	51,53%	49,83%	46,94%	44,80%
47	7	5,51%	23,65%	20,49%	20,80%	21,91%	20,94%
48	7	5,41%	21,21%	17,98%	17,20%	18,18%	18,64%

RESULTADOS DO ENSAIO DA HIPÓTESE 2

Tabela 13: Eficácia da detecção do ensaio da hipótese 2

Vídeos	Cenários	Tratamentos		
		Original	HD	Half FHD
1	1	78,96%	84,12%	78,71%
2	1	84,59%	87,24%	85,83%
3	1	5,68%	5,84%	5,29%
4	1	17,32%	18,99%	14,37%
5	1	29,28%	29,93%	29,35%
6	2	95,64%	93,01%	93,54%
7	2	96,81%	95,70%	95,99%
8	4	12,62%	13,40%	11,77%
9	4	20,22%	17,06%	19,38%
10	4	12,18%	14,07%	12,46%
11	4	17,75%	21,42%	15,89%
12	4	48,94%	48,04%	40,35%
13	4	19,04%	18,28%	14,89%
14	4	17,43%	17,68%	14,99%
15	4	33,81%	35,01%	31,45%
16	4	23,80%	25,09%	22,67%
17	4	42,39%	43,71%	39,17%

Vídeos	Cenários	Tratamentos		
		Original	HD	Half FHD
18	4	40,15%	33,81%	32,82%
19	4	40,89%	36,98%	37,47%
20	4	39,68%	35,00%	30,79%
21	4	23,39%	24,85%	23,60%
22	4	14,63%	14,59%	11,84%
23	4	13,17%	13,63%	11,89%
24	5	32,63%	30,37%	32,16%
25	5	47,10%	43,83%	45,57%
26	5	31,10%	27,98%	31,83%
27	5	66,03%	58,90%	65,13%
28	5	40,53%	36,91%	35,10%
29	5	6,39%	5,83%	5,75%
30	5	58,87%	56,62%	57,65%
31	5	55,00%	50,69%	49,29%
32	5	21,52%	22,31%	21,95%
33	5	30,08%	26,77%	27,17%
34	5	12,89%	12,57%	12,70%
35	5	53,31%	53,12%	52,34%
36	5	24,34%	22,91%	24,05%
37	5	15,67%	14,78%	14,98%
38	6	12,63%	11,57%	11,74%
39	6	11,54%	7,37%	7,86%
40	6	14,25%	10,79%	12,01%
41	6	26,81%	22,34%	22,14%
42	6	24,37%	21,12%	23,12%
43	7	29,20%	24,71%	26,27%
44	7	29,09%	24,87%	24,84%

Vídeos	Cenários	Tratamentos		
		Original	HD	Half FHD
45	7	19,84%	10,78%	12,11%
46	7	54,20%	51,32%	41,68%
47	7	23,65%	20,06%	17,35%
48	7	21,21%	17,80%	16,28%

RESULTADOS DO ENSAIO DA HIPÓTESE 3

Tabela 14: Eficácia da detecção do ensaio da hipótese 3

Vídeos	Cenários	Tratamentos			
		Original	HD@5 kbps	HD@4 kbps	HD@5 kbps
1	1	78,96%	80,11%	79,46%	79,61%
2	1	84,59%	84,25%	83,55%	82,76%
3	1	5,68%	5,61%	5,61%	5,45%
4	1	17,32%	17,01%	16,82%	16,43%
5	1	29,28%	27,04%	26,48%	25,95%
6	2	95,64%	91,15%	91,05%	91,01%
7	2	96,81%	94,00%	93,93%	93,03%
8	4	12,62%	13,81%	13,75%	15,02%
9	4	20,22%	15,32%	17,76%	15,21%
10	4	12,18%	15,24%	15,97%	15,57%
11	4	17,75%	22,04%	21,29%	22,16%
12	4	48,94%	44,22%	45,08%	44,81%
13	4	19,04%	18,38%	18,63%	18,48%
14	4	17,43%	19,27%	18,20%	19,42%
15	4	33,81%	35,25%	34,53%	33,90%
16	4	23,80%	26,26%	25,65%	25,88%
17	4	42,39%	44,92%	46,39%	45,94%

Vídeos	Cenários	Tratamentos			
		Original	HD@5 kbps	HD@4 kbps	HD@5 kbps
18	4	40,15%	33,46%	33,93%	35,20%
19	4	40,89%	34,82%	34,68%	35,25%
20	4	39,68%	34,45%	33,70%	34,41%
21	4	23,39%	22,62%	22,97%	22,78%
22	4	14,63%	13,34%	13,15%	13,78%
23	4	13,17%	11,63%	11,71%	12,60%
24	5	32,63%	29,05%	28,22%	28,78%
25	5	47,10%	42,27%	41,87%	42,97%
26	5	31,10%	26,39%	26,69%	25,97%
27	5	66,03%	59,43%	57,91%	58,81%
28	5	40,53%	29,14%	30,01%	32,32%
29	5	6,39%	5,50%	5,30%	4,80%
30	5	58,87%	53,89%	54,83%	54,98%
31	5	55,00%	48,95%	46,97%	48,44%
32	5	21,52%	21,63%	20,97%	20,76%
33	5	30,08%	24,27%	24,77%	23,88%
34	5	12,89%	11,26%	11,14%	11,52%
35	5	53,31%	53,45%	54,19%	53,10%
36	5	24,34%	21,29%	21,36%	20,25%
37	5	15,67%	12,91%	12,00%	12,22%
38	6	12,63%	9,85%	10,09%	10,13%
39	6	11,54%	5,83%	5,84%	5,20%
40	6	14,25%	8,51%	8,59%	8,77%
41	6	26,81%	21,20%	20,19%	20,70%
42	6	24,37%	19,10%	18,27%	19,71%
43	7	29,20%	24,93%	24,20%	22,10%
44	7	29,09%	23,39%	22,71%	24,98%

Vídeos	Cenários	Tratamentos			
		Original	HD@5 kbps	HD@4 kbps	HD@5 kbps
45	7	19,84%	8,62%	7,53%	8,55%
46	7	54,20%	47,49%	49,81%	43,36%
47	7	23,65%	17,84%	17,87%	17,68%
48	7	21,21%	15,45%	15,90%	16,08%

APÊNDICE C: CÓDIGOS FONTES

DETECTOR HOG

Software responsável por detectar pessoas nos fluxos de vídeos. Utiliza a implementação do OpenCV (BRADSKI, 2000) do HOG (DALAL; TRIGGS, 2005). Recebe como entrada o fluxo de vídeo a ser processado, e opcionalmente, a resolução espacial que para a qual o fluxo de vídeo deve ser reduzido antes do processamento. As saídas são impressas na saída padrão com as detecções identificadas, sendo o primeiro campo o índice do quadro da detecção, seguido da informação do retângulo em que a detecção foi realizada e por fim de um campo que indica o grau de certeza na detecção pelo detector.

Código fonte 1: Detector de objetos HOG.

```
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/objdetect/objdetect.hpp>
#include <opencv2/contrib/contrib.hpp>
#include <fstream>
#include <iostream>
#include <ctime>
#include <algorithm>

using namespace cv;
using namespace std;

/*
 * Detector
 * O primeiro argumento é o arquivo de vídeo
 * O segundo argumento, opcional, é o redimensionamento que deve ser feito antes da
 * detecção.
 */
int main(int argc, char** argv) {
    VideoCapture capture(argv[1]);
    int preliminarResize = 0, upscaleBefore = 0, w = 480, h = 360;
    if (argc > 2) {
```

```

preliminarResize = atoi(argv[2]);
upscaleBefore = 1;
switch (preliminarResize) {
    case 1:
        w = 1280;
        h = 720;
        break;
    case 2:
        w = 960;
        h = 540;
        break;
    case 3:
        w = 540;
        h = 360;
        break;
}
}
if (capture.get(CV_CAP_PROP_FRAME_HEIGHT) != 1080) {
    upscaleBefore = 1;
}
Mat frame;
HOGDescriptor hog;
hog.setSVMDetector(HOGDescriptor::getDefaultPeopleDetector());
long frame_count = 0;
while (capture.read(frame)) {
    // Redimensiona para menos
    if (preliminarResize) {
        Mat temp;
        resize(frame, temp, Size2i(w, h), 0, 0, INTER_AREA);
        frame = temp;
    }
    // Retorna para o tamanho original (FullHD)
    if (upscaleBefore) {
        Mat temp;

```

```

        resize(frame, temp, Size2i(1920, 1080), 0, 0, INTER_AREA);
        frame = temp;
    }
    // Detecção
    vector<Rect> found;
    vector<double> weights;
    hog.detectMultiScale(frame, found, weights, 0, Size(8,8), Size(24,16), 1.05, -
9999, false);
    for (size_t i = 0; i < found.size(); i++) {
        cout << frame_count << "\t" << max(found[i].x, 0) << "\t" <<
max(found[i].y, 0);
        cout << "\t" << min(found[i].x+found[i].width, 1920) << "\t" <<
min(found[i].y+found[i].height, 1080);
        cout << "\t" << weights[i] << endl;
    }
    frame_count++;
}
capture.release();
return 0;
}

```

CÁLCULO DA MÉTRICA DE EFICÁCIA

Software que produz a métrica AUC-PR considerando todos os quadros do fluxo de vídeo. Desenvolvido em Python (ROSSUM; DRAKE, 2001) recebe como entrada um diretório contendo os arquivos de objetos identificados na base VIRAT (OH et al., 2011) e um diretório com os arquivos de detecções identificados pelo HOG (DALAL; TRIGGS, 2005). Para cada par de arquivos de objeto da base VIRAT e de detecções do HOG é produzido a métrica AP e gerado um gráfico precisão e revocação.

Código fonte 2: Cálculo da métrica de eficácia.

```

import sys
from sklearn.metrics import auc
import numpy as np
import pylab as pl

```

```

import os

# Realiza o calculo de percentual da area de intersecção de dois retangulos com a area total
dos retangulos
def calc_spatial_match(detections, ground_truth):
    rx, ry, rw, rh = detections[:4]
    qx, qy, qw, qh = ground_truth
    rw -= rx
    rh -= ry

    intersect_area = max(0, min(rx+rw,qx+qw) - max(rx,qx))*max(0, min(ry+rh,qy+qh) -
max(ry,qy))
    union_area = rw*rh+qw*qh-intersect_area
    return float(intersect_area)/union_area

# Leitura do arquivo de objetos do VIRAT
def read_layout_virat(f):
    fgt = file(f, 'r')
    m = {}
    line = fgt.readline()
    while line != "":
        line = line.split()
        if ('1' == line[7]):
            #id = int(line[0])
            current = int(line[2])
            r = (int(line[3]),int(line[4]),int(line[5]),int(line[6]))
            if (not m.has_key(current)):
                m[current] = []
            m[current].append(r)
        line = fgt.readline()
    fgt.close()
    return m

# Leitura do arquivo de deteccoes do HOG

```

```

def read_detection(d, fs):
    m = {}
    for f in fs:
        fgt = file(d+os.path.sep+f, 'r')
        line = fgt.readline()
        while line != "":
            line = line.split("\t")
            frame = int(line[0])
            r = [int(line[1]),int(line[2]),int(line[3]),int(line[4]),float(line[5])]
            if (not m.has_key(frame)):
                m[frame] = list()
                m[frame].append(r)
            line = fgt.readline()
        fgt.close()
    return m

```

```

# leitura dos arquivos VIRAT
outdir = (sys.argv[3]+"/") if (len(sys.argv) > 3) else ""
files = sorted([f for f in os.listdir(sys.argv[1])])
ca = 0
th = 0.1
for fgt in files:
    detections = []
    undetections = 0
    arq = fgt[0:fgt.index('.')]
    fr = sorted([f for f in os.listdir(sys.argv[2]) if (arq in f)])
    # leitura dos arquivos do virat
    gt = read_layout_virat(sys.argv[1]+os.path.sep+fgt)
    if (len(fr) > 0) and (len(gt.keys()) > 0):
        ca += 1
        # leitura dos arquivos de deteccion
        treats = read_detection(sys.argv[2], fr)
        max_frame = [max(sorted(gt.keys()))]
        max_frame.extend([max(sorted(treats.keys()))])

```

```

max_frame = max(max_frame)
# identificando positivos e falsos
for frame_i in range(max_frame):
    for d in treats.get(frame_i, []):
        d.append(0.0)
    for dgt in gt.get(frame_i, []):
        d[len(d)-1] = max(d[len(d)-1], calc_spatial_match(d, dgt))
    detections.append((1 if d[5] >= th else 0, d[4]))
    for dgt in gt.get(frame_i, []):
        match = 0
        for d in treats.get(frame_i, []):
            match = max(match, calc_spatial_match(d, dgt))
        if (match < th) and (dgt[3] >= 96):
            undetections += 1.
detections = np.array(detections)
np.argsort(detections[:, 1])
# calculando metricas
ttp = 0
for d in detections:
    if (d[0] == 1):
        ttp += 1.
ctp = 0
cfp = 0
recall = []
precision = []
for i,d in enumerate(detections):
    if (d[0] == 1):
        ctp += 1.
    else:
        cfp += 1.
    precision.append((ctp / max(ctp + cfp, 1)))
    recall.append((ctp / max(ttp + undetections, 1)))
recall, precision = (list(t) for t in zip(*sorted(zip(recall, precision))))
precision = np.array(precision)

```

```

recall = np.array(recall)
area = auc(recall, precision)
# gerando grafico
pl.clf()
pl.plot(recall, precision, label='PR curve')
pl.xlabel('Recall')
pl.ylabel('Precision')
pl.ylim([0.0, 1.05])
pl.xlim([0.0, 1.05])
pl.title(arq+ ': AUC=%f' % area)
pl.legend(loc="upper right")
pl.savefig(outdir+arq+'.png', bbox_inches=0)
with open(outdir+'result.txt', 'a') as f:
    f.write(arq + '\t' + str(area) + '\n')
print arq + '\t' + str(area)

```

CÁLCULOS ESTATÍSTICOS

Software que realiza os testes estatísticos de Friedman e Wilconxon e produz os gráficos utilizados neste trabalho. Desenvolvido em linguagem R (R CORE TEAM, 2014) recebe como entrada os arquivos contendo a métrica AP para cada fluxo de vídeo nos diferentes níveis de compressão e resolução espacial e produz como saída os resultados dos testes estatísticos e os gráficos comparativos.

Código fonte 3: Cálculo estatístico.

```

library(nortest) # Testes de normalidade
library(readODS)
library(reshape)
library(plyr)
library(bear)
library(ggplot2)

planilha = read.ods(file = "niveis.ods", 1)

planilha2 = planilha[-1,]

```

```
planilha2 = planilha2[,-1]
rownames(planilha2) <- planilha[-1,1]
colnames(planilha2) <- planilha[1,-1]
planilha = planilha2
rm(planilha2)
planilha = as.data.frame(gsub(",", ".", as.matrix(planilha)))

#bitrate = read.ods("/home/marcell/mestrado/experiments/data/results/niveis.ods", 3)
bitrate = read.ods(file = "I:/Dados/Mestrado/experiments/data/results/niveis.ods", 3)

planilha2 = bitrate[-1,]
planilha2 = planilha2[,-1]
planilha2 = as.data.frame(gsub(",", ".", as.matrix(planilha2)))
rownames(planilha2) <- bitrate[-1,1]
colnames(planilha2) <- bitrate[1,-1]
bitrate = planilha2
rm(planilha2)

qamostras = length(planilha[,1])
original <- as.numeric(as.vector(planilha$Original))

## Teste de normalidade

shapiro.test(original)
ad.test(original)
lillie.test(original)

## Hipotese 1

v5k <- as.numeric(as.vector(planilha$"5kbps"))
v4k <- as.numeric(as.vector(planilha$"4kbps"))
v3k <- as.numeric(as.vector(planilha$"3kbps"))
v2k <- as.numeric(as.vector(planilha$"2kbps"))
```



```

compressao <- matrix(c(original,v5k,v4k,v3k,v2k), nrow = qamostras, byrow = FALSE,
                    dimnames = list(1 : qamostras, c("Original", "5", "4", "3", "2")))
friedman.test(compressao)

```

```

postos = matrix(nrow = qamostras, ncol=5, byrow = FALSE,
                dimnames = list(1 : qamostras, c("Original", "5", "4", "3", "2")))

```

```

for(i in 1:dim(compressao)[1]) {
  postos[i,] = order(compressao[i,])
}

```

```
ROriginal1 = sum(postos[,1])
```

```
R5kbps = sum(postos[,2])
```

```
R4kbps = sum(postos[,3])
```

```
R3kbps = sum(postos[,4])
```

```
R2kbps = sum(postos[,5])
```

```
q = 2.16; N = qamostras; k = 5
```

```
est1 = q * sqrt((N * k * (k+1)) / 6)
```

```
abs(ROriginal1-R5kbps) >= est1
```

```
abs(ROriginal1-R4kbps) >= est1
```

```
abs(ROriginal1-R3kbps) >= est1
```

```
abs(ROriginal1-R2kbps) >= est1
```

```
boxplot(compressao, ylab="Eficácia (AUC-PR)", xlab="Taxa de transmissão (kbps)")
```

```
## Hipotese 2
```

```
vHD <- as.numeric(as.vector(planilha$"Original@HD"))
```

```
vhFHD <- as.numeric(as.vector(planilha$"Original@HFHD"))
```

```

resolucao <-matrix(c(original, vHD, vhFHD), nrow = qamostras, byrow = FALSE,
                  dimnames = list(1 : qamostras, c("Full HD", "HD", "Half Full HD")))

```

```
friedman.test(resolucao)
```

```
postos = matrix(nrow = qamostras, ncol=3, byrow = FALSE,
               dimnames = list(1 : qamostras, c("FullHD", "HD", "HFHD")))
```

```
for(i in 1:dim(resolucao)[1]) {
  postos[i,] = order(resolucao[i,])
}
```

```
ROriginal2 = sum(postos[,1])
```

```
RHD = sum(postos[,2])
```

```
RHFHD = sum(postos[,3])
```

```
q=1.92; N = qamostras; k = 3
```

```
est2 = q * sqrt((N * k * (k+1)) / 6)
```

```
abs(ROriginal2-RHD) >= est2
```

```
abs(ROriginal2-RHFHD) >= est2
```

```
boxplot(resolucao, ylab="Eficácia (AUC-PR)", xlab="Resolução")
```

```
## Hipotese 3
```

```
v5kHD <- as.numeric(as.vector(planilha$"5kbps@HD"))
```

```
v4kHD <- as.numeric(as.vector(planilha$"4kbps@HD"))
```

```
v3kHD <- as.numeric(as.vector(planilha$"3kbps@HD"))
```

```
comp_res <- matrix(c(original, v5kHD, v4kHD, v3kHD), nrow = qamostras, byrow =
FALSE,
```

```
dimnames = list(1 : qamostras, c("Original", "5@HD", "4@HD", "3@HD")))
```

```
friedman.test(comp_res)
```

```
postos = matrix(nrow = qamostras, ncol=4, byrow = FALSE,
```

```

dimnames = list(1 : qamostras, c("Original", "5kbps@HD", "4kbps@HD",
"3kbps@HD"))

for(i in 1:dim(comp_res)[1]){
  postos[i,] = order(comp_res[i,])
}

ROriginal3 = sum(postos[,1])
R5kbps = sum(postos[,2])
R4kbps = sum(postos[,3])
R3kbps = sum(postos[,4])

q = 2.06; N = qamostras; k = 4
est3 = q*sqrt((N * k * (k+1))/6)

abs(ROriginal3-R5kbps) >= est3
abs(ROriginal3-R4kbps) >= est3
abs(ROriginal3-R3kbps) >= est3

boxplot(comp_res, ylab="Eficácia (AUC-PR)", xlab="Taxa de transmissão (kbps) @
Resolução")

# Análise Adicional

t1 = data.frame(original)
t1[, "Resolucao"] = 1
t1[, "Taxa de bits"] = "Original"
t2 = data.frame(v5k)
t2[, "Resolucao"] = 1
t2[, "Taxa de bits"] = 5
t3 = data.frame(v4k)
t3[, "Resolucao"] = 1
t3[, "Taxa de bits"] = 4
t4 = data.frame(v3k)

```

```

t4[,"Resolucao"] = 1
t4[,"Taxa de bits"] = 3
t5 = data.frame(v2k)
t5[,"Resolucao"] = 1
t5[,"Taxa de bits"] = 2
t6 = data.frame(v5kHD)
t6[,"Resolucao"] = 2
t6[,"Taxa de bits"] = 5
t7 = data.frame(v4kHD)
t7[,"Resolucao"] = 2
t7[,"Taxa de bits"] = 4
t8 = data.frame(v3kHD)
t8[,"Resolucao"] = 2
t8[,"Taxa de bits"] = 3
colnames(t1) = c("Acuracia", "Resolucao", "Bitrate")
colnames(t2) = c("Acuracia", "Resolucao", "Bitrate")
colnames(t3) = c("Acuracia", "Resolucao", "Bitrate")
colnames(t4) = c("Acuracia", "Resolucao", "Bitrate")
colnames(t5) = c("Acuracia", "Resolucao", "Bitrate")
colnames(t6) = c("Acuracia", "Resolucao", "Bitrate")
colnames(t7) = c("Acuracia", "Resolucao", "Bitrate")
colnames(t8) = c("Acuracia", "Resolucao", "Bitrate")
values = rbind(t1,t2,t3,t4,t5,t6,t7,t8)

dfc <- summarySE(values, measurevar="Acuracia", groupvars=c("Resolucao","Bitrate"))
dfc[1:4,] = dfc[4:1,]
dfc[6:8,] = dfc[8:6,]
dfc[1:5,1] = "1"
dfc[6:8,1] = "2"

avg = mean(as.numeric(as.vector(bitrate$Bitrate)))

# grafico que une os diferentes resultados das hipoteses
pd <- position_dodge(.1)

```

```

ggplot(dfc, aes(x=Bitrate, y=Acuracia, colour=Resolucao, group=Resolucao)) +
  geom_errorbar(aes(ymin=Acuracia-ci, ymax=Acuracia+ci), width=.1, position=pd) +
  geom_line(position=pd) +
  geom_point(position=pd, size=3, shape=21, fill="black") +
  scale_x_discrete(limits=c("Original",5,4,3,2), breaks=c("Original",5,4,3,2), labels =
c("Original",paste(trunc(500000/avg),"% / 5"), paste(trunc(400000/avg),"% /
4"),paste(trunc(300000/avg),"% / 3"),paste(trunc(200000/avg),"% / 2"))) +
  xlab("Taxa de transmissão (% e kbps)") +
  ylab("Eficácia (AUC-PR)") +
  scale_colour_hue(name="Resolução", breaks=c("1", "2"), labels=c("Full HD", "HD"), l=40)
+
  theme_bw() +
  theme(legend.position="right")

# grafico de compressao

boriginal <- as.numeric(as.vector(bitrate$Bitrate))
bcen4 <- as.numeric(as.vector(bitrate$Bitrate[8:23]))
bcen5 <- as.numeric(as.vector(bitrate$Bitrate[24:37]))

cbind.fill <- function(...){
  nm <- list(...)
  nm <- lapply(nm, as.matrix)
  n <- max(sapply(nm, nrow))
  do.call(cbind, lapply(nm, function (x)
    rbind(x, matrix(, n-nrow(x), ncol(x)))))
}

rm(br)
br <- cbind.fill(data.frame(boriginal),data.frame(bcen4),data.frame(bcen5))
br = br/1000

boxplot(br, names = c("Todos","4","5"), ylab="Taxa de transmissão (kbps)",
xlab="Cenários")

```

```

## Cenário 4
planilha = read.ods(file = "niveis.ods", 1)

planilha2 = planilha[-1,]
planilha2 = planilha2[,-1]
rownames(planilha2) <- planilha[-1,1]
colnames(planilha2) <- planilha[1,-1]
planilha = planilha2
rm(planilha2)
planilha = as.data.frame(gsub(",", ".", as.matrix(planilha)))
planilha = planilha[8:23,]

qamostras = length(planilha[,1])
## Hipotese 1

original <- as.numeric(as.vector(planilha$Original))
v5k <- as.numeric(as.vector(planilha$"5kbps"))
v4k <- as.numeric(as.vector(planilha$"4kbps"))
v3k <- as.numeric(as.vector(planilha$"3kbps"))
v2k <- as.numeric(as.vector(planilha$"2kbps"))

compressao <- matrix(c(original,v5k,v4k,v3k,v2k), nrow = qamostras, byrow = FALSE,
                    dimnames = list(1 : qamostras, c("Original", "5", "4", "3", "2")))
friedman.test(compressao)

postos = matrix(nrow = qamostras, ncol=5, byrow = FALSE,
               dimnames = list(1 : qamostras, c("Original", "5", "4", "3", "2")))

for(i in 1:dim(compressao)[1]) {
  postos[i,] = order(compressao[i,])
}

ROriginal1 = sum(postos[,1])

```

```

R5kbps = sum(postos[,2])
R4kbps = sum(postos[,3])
R3kbps = sum(postos[,4])
R2kbps = sum(postos[,5])

q = 2.16; N = qamostras; k = 5
est1 = q * sqrt((N * k * (k+1)) / 6)

abs(ROriginal1-R5kbps) >= est1
abs(ROriginal1-R4kbps) >= est1
abs(ROriginal1-R3kbps) >= est1
abs(ROriginal1-R2kbps) >= est1

## Hipotese 2

vHD <- as.numeric(as.vector(planilha$"Original@HD"))
vhFHD <- as.numeric(as.vector(planilha$"Original@HFHD"))

resolucao <-matrix(c(original, vHD, vhFHD), nrow = qamostras, byrow = FALSE,
                  dimnames = list(1 : qamostras, c("Full HD", "HD", "Half Full HD")))
friedman.test(resolucao)

postos = matrix(nrow = qamostras, ncol=3, byrow = FALSE,
               dimnames = list(1 : qamostras, c("FullHD", "HD", "HFHD")))

for(i in 1:dim(resolucao)[1]) {
  postos[i,] = order(resolucao[i,])
}

ROriginal2 = sum(postos[,1])
RHHD = sum(postos[,2])
RHFHD = sum(postos[,3])

q=1.92; N = qamostras; k = 3

```

```
est2 = q * sqrt((N * k * (k+1)) / 6)
```

```
abs(ROriginal2-RHD) >= est2
```

```
abs(ROriginal2-RHFHD) >= est2
```

```
## Hipotese 3
```

```
v5kHD <- as.numeric(as.vector(planilha$"5kbps@HD"))
```

```
v4kHD <- as.numeric(as.vector(planilha$"4kbps@HD"))
```

```
v3kHD <- as.numeric(as.vector(planilha$"3kbps@HD"))
```

```
comp_res <- matrix(c(original, v5kHD, v4kHD, v3kHD), nrow = qamostras, byrow = FALSE,
```

```
                  dimnames = list(1 : qamostras, c("Original", "5@HD", "4@HD", "3@HD")))
```

```
friedman.test(comp_res)
```

```
postos = matrix(nrow = qamostras, ncol=4, byrow = FALSE,
```

```
                  dimnames = list(1 : qamostras, c("Original", "5kbps@HD", "4kbps@HD", "3kbps@HD")))
```

```
for(i in 1:dim(comp_res)[1]){
```

```
  postos[i,] = order(comp_res[i,])
```

```
}
```

```
ROriginal3 = sum(postos[,1])
```

```
R5kbps = sum(postos[,2])
```

```
R4kbps = sum(postos[,3])
```

```
R3kbps = sum(postos[,4])
```

```
q = 2.06; N = qamostras; k = 4
```

```
est3 = q*sqrt((N * k * (k+1))/6)
```

```
abs(ROriginal3-R5kbps) >= est3
```

```
abs(ROriginal3-R4kbps) >= est3
```



```
abs(ROriginal3-R3kbps) >= est3
```

```
## Cenário 5
```

```
planilha = read.ods(file = "niveis.ods", 1)
```

```
planilha2 = planilha[-1,]
```

```
planilha2 = planilha2[,-1]
```

```
rownames(planilha2) <- planilha[-1,1]
```

```
colnames(planilha2) <- planilha[1,-1]
```

```
planilha = planilha2
```

```
rm(planilha2)
```

```
planilha = as.data.frame(gsub(",", ".", as.matrix(planilha)))
```

```
planilha = planilha[24:37,]
```

```
## Hipotese 1
```

```
original <- as.numeric(as.vector(planilha$Original))
```

```
v5k <- as.numeric(as.vector(planilha$"5kbps"))
```

```
v4k <- as.numeric(as.vector(planilha$"4kbps"))
```

```
v3k <- as.numeric(as.vector(planilha$"3kbps"))
```

```
v2k <- as.numeric(as.vector(planilha$"2kbps"))
```

```
compressao <- matrix(c(original,v5k,v4k,v3k,v2k), nrow = qamostras, byrow = FALSE,
```

```
dimnames = list(1 : qamostras, c("Original", "5", "4", "3", "2")))
```

```
friedman.test(compressao)
```

```
postos = matrix(nrow = qamostras, ncol=5, byrow = FALSE,
```

```
dimnames = list(1 : qamostras, c("Original", "5", "4", "3", "2")))
```

```
for(i in 1:dim(compressao)[1]) {
```

```
  postos[i,] = order(compressao[i,])
```

```
}
```

```

ROriginal1 = sum(postos[,1])
R5kbps = sum(postos[,2])
R4kbps = sum(postos[,3])
R3kbps = sum(postos[,4])
R2kbps = sum(postos[,5])

q = 2.16; N = qamostras; k = 5
est1 = q * sqrt((N * k * (k+1)) / 6)

abs(ROriginal1-R5kbps) >= est1
abs(ROriginal1-R4kbps) >= est1
abs(ROriginal1-R3kbps) >= est1
abs(ROriginal1-R2kbps) >= est1

## Hipotese 2

vHD <- as.numeric(as.vector(planilha$"Original@HD"))
vhFHD <- as.numeric(as.vector(planilha$"Original@HFHD"))

resolucao <-matrix(c(original, vHD, vhFHD), nrow = qamostras, byrow = FALSE,
                  dimnames = list(1 : qamostras, c("Full HD", "HD", "Half Full HD")))
friedman.test(resolucao)

postos = matrix(nrow = qamostras, ncol=3, byrow = FALSE,
                dimnames = list(1 : qamostras, c("FullHD", "HD", "HFHD")))

for(i in 1:dim(resolucao)[1]) {
  postos[i,] = order(resolucao[i,])
}

ROriginal2 = sum(postos[,1])
RHHD = sum(postos[,2])
RHFHD = sum(postos[,3])

```

```
q=1.92; N = qamostras; k = 3
est2 = q * sqrt((N * k * (k+1)) / 6)
```

```
abs(ROriginal2-RHD) >= est2
abs(ROriginal2-RHFHD) >= est2
```

```
## Hipotese 3
```

```
v5kHD <- as.numeric(as.vector(planilha$"5kbps@HD"))
v4kHD <- as.numeric(as.vector(planilha$"4kbps@HD"))
v3kHD <- as.numeric(as.vector(planilha$"3kbps@HD"))
```

```
comp_res <- matrix(c(original, v5kHD, v4kHD, v3kHD), nrow = qamostras, byrow =
FALSE,
```

```
dimnames = list(1 : qamostras, c("Original", "5@HD", "4@HD", "3@HD")))
friedman.test(comp_res)
```

```
postos = matrix(nrow = qamostras, ncol=4, byrow = FALSE,
```

```
dimnames = list(1 : qamostras, c("Original", "5kbps@HD", "4kbps@HD",
"3kbps@HD")))

```

```
for(i in 1:dim(comp_res)[1]){
  postos[i,] = order(comp_res[i,])
}
```

```
ROriginal3 = sum(postos[,1])
```

```
R5kbps = sum(postos[,2])
```

```
R4kbps = sum(postos[,3])
```

```
R3kbps = sum(postos[,4])
```

```
q = 2.06; N = qamostras; k = 4
```

```
est3 = q*sqrt((N * k * (k+1))/6)
```

```
abs(ROriginal3-R5kbps) >= est3
```

$\text{abs}(\text{ROriginal3}-\text{R4kbps}) \geq \text{est3}$

$\text{abs}(\text{ROriginal3}-\text{R3kbps}) \geq \text{est3}$