Teorema[chapter] *Demonstração.*

□

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS CORNÉLIO PROCÓPIO
DIRETORIA DE GRADUAÇÃO E EDUCAÇÃO PROFISSIONAL
DEPARTAMENTO DE COMPUTAÇÃO
ENGENHARIA DE COMPUTAÇÃO

MIKE PATRICK MERCANTE

ESTUDO DO IMPACTO DA RESOLUÇÃO DE IMAGENS E CUSTO COMPUTACIONAL NA SEGMENTAÇÃO DE VÍDEOS

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO
2017

Minstério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Cornélio Procópio
Diretoria de Graduação e Educação Profissional
Departamento de Computação
Engenharia de Computação

TERMO DE APROVAÇÃO

Estudo do impacto da resolução de imagens e custo computacional na segmentação de vídeos

por

Mike Patrick Mercante

Esta Trabalho de Conclusão de Curso foi julgada adequada para obtenção do Título de "Bacharel em Engenharia de Computação" e aprovado em sua forma final pelo Departamento de Computação da Universidade Tecnológica Federal do Paraná.
Cornélio Procópio, 10/11/2017.

_____

,
do Curso

Banca Examinadora:

_____

Silvio R. R. Sanches, Prof. Dr.
Orientador

_____

Primeiro Membro da Banca, Título
Universidade

_____

Segundo Membro da Banca, Título
Universidade

"A Folha de Aprovação assinada encontra-se na Coordenação do Curso do Programa"

RESUMO

MERCANTE, Mike Patrick. **Estudo do impacto da resolução de imagens e custo computacional na segmentação de vídeos**. 2017. 32 f. Trabalho de Conclusão de Curso – Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2017.

Na área da Visão Computacional, a segmentação de imagens para extração de elementos de interesse têm sido vastamente pesquisada. Entre as diversas aplicações, podem ser citados sistemas de videoconferências, Realidade Aumentada e substituição de fundo em uma cena. No entanto, aplicações que tratam imagens capturadas por câmeras de segurança se destacam pela sua importância, seja em ambientes domésticos, comerciais ou industriais. No decorrer do tempo, vários métodos foram desenvolvidos com o objetivo de otimizar o processo de segmentação, aumentando sua acurácia e diminuindo o tempo de processamento. Dependendo do tipo de aplicação, seja em ambientes abertos ou fechados, com variância de luz ou não, um método é mais recomendado. O objetivo deste trabalho é realizar um estudo em imagens de resolução 4k (3840 x 2160) e resoluções intermediárias, analisando o desempenho de um algoritmo de segmentação de objeto de interesse em termos de custo computacional e acurácia. Para atingir este objetivo foi gravado um vídeo em 4k de uma cena típica de cenários de segurança por vídeo, processou-se essas imagens com um algoritmo escolhido de modo a ser possível serem obtidas métricas de acurácia e desempenho em relação ao custo computacional gasto.

Palavras-chave: Alta resolução. Segmentação de video. Custo computacional.

ABSTRACT

MERCANTE, Mike Patrick. **Study of the impact of image resolution and computational cost on video segmentation.** 2017. 32 f. Undergraduate Thesis – Computer Engineering, Federal University of Technology - Paraná. Cornélio Procópio, 2017.

In the area of ??Computational Vision, the segmentation of images to extract elements of interest have been extensively researched. Among the various applications, videoconference systems, Augmented Reality and background substitution in a scene can be mentioned. However, applications that treat images captured by security cameras stand out for their importance, whether in domestic, commercial or industrial environments. Over time, several methods have been developed with the objective of optimizing the segmentation process, increasing its accuracy and reducing processing time. Depending on the type of application, whether indoors or outdoors, with light variance or not, a method is most recommended. The objective of this work is to perform a study on 4k resolution images (3840 x 2160) and intermediate resolutions, analyzing the performance of an object segmentation algorithm of interest in terms of computational cost and accuracy. To achieve this goal a 4k video of a typical scene of video security scenarios was recorded, these images were processed with a chosen algorithm in order to be able to obtain metrics of accuracy and performance in relation to the computational cost spent.

Keywords: High Resolution. Video Segmentation. Computacional cost.

LISTA DE ILUSTRAÇÕES

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE ALGORITMOS

SUMÁRIO

1 INTRODUÇÃO

Sistemas computacionais de segurança se tornaram uma ferramenta muito utilizada, seja em ambientes domésticos, comerciais ou industriais. A análise das imagens de vídeos, muitas vezes realizada em tempo real, capturadas nesses espaços pode aumentar a segurança, detectando comportamentos anormais, quedas de pessoas, acidentes automobilísticos etc (**??**). A aplicação de técnicas visão computacional para a extração de elementos de interesse do conteúdo dos quadros desses vídeos é uma tarefa necessária nessas aplicações (**??**).

Outros exemplos de aplicações que exigem extração de elementos de interesse são os sistemas de videoconferências com substituição de fundo, Realidade Aumentada, além de substituição de fundo em aplicações como na previsão do tempo em telejornais. Nestas aplicações, no entanto, o ambiente geralmente é controlado, pois possui iluminação constante, fundo monocromático e nenhuma movimentação dos demais elementos que compõem a cena.

Nas aplicações voltadas para segurança por vídeo, o fundo da cena é arbitrário e, em consequência disso, a extração dos elementos de interesse deve ser realizada por algoritmos que atuem nessas condições. Algoritmos desse tipo se apoiam em informações como contraste entre *pixels* vizinhos e coerência temporal entre quadros consecutivos. Algumas abordagens consideram, inclusive, o conhecimento prévio da forma do elemento de interesse para identificar os *pixels* que pertencem ao fundo e os que pertencem ao próprio elemento.

O objetivo deste trabalho de conclusão de curso é analisar a relação performance x recurso computacional em imagens de alta resolução. Para isto foi escolhido um algoritmo ranqueado no site changedetection[1], site este dedicado à ranquear algoritmos que lidam com processamento de imagem.

Como imagens de alta resolução sendo cada vez mais acessíveis, a avaliação consiste em prover um estudo que relate qual é a relação entre “desempenho do algoritmo x resolução da imagem x custo computacional”.

1.1 MOTIVAÇÃO

A extração de elementos de interesse, em tempo real, de quadros de vídeos é uma tarefa necessária para um número significativo de aplicações, muitas delas voltadas para segurança por vídeo. Os algoritmos mais robustos encontrados na literatura utilizam vários tipos de informações, como cor, contraste e coerência temporal entre quadros consecutivos para estimar se o pixel processado pertence ao elemento de interesse ou ao fundo. Algumas abordagens consideram, inclusive, o conhecimento prévio da forma do elemento a ser extraído ou valores de profundidade

[1] Changedetection.net

de pixels (obtidas de sensores) para auxiliar a segmentação.

Conjuntos de dados disponíveis online[2] utilizam imagens capturadas por câmeras de baixa resolução, porém oferecem uma boa variedade de cenários que simulam aplicações em situações reais. Porém, visto que câmeras capazes de capturar imagens em alta resolução vem se popularizando, se mostra relevante o estudo de se a resolução impacta no desempenho do algoritmo, e ainda mais, se o resultado final justifica o custo computacional adicional requerido.

1.2 OBJETIVOS

Os objetivos da presente pesquisa são apresentados em duas subseções. A primeira delas traz o objetivo geral do trabalho e a segunda detalha as metas a serem atingidas (objetivos específicos) para que tal objetivo seja alcançado.

1.2.1 Objetivo Geral

O objetivo deste presente trabalho é verificar se ao processar um vídeo, em várias resoluções, há ganho na acurácia da extração do objeto de interesse em imagens de alta resolução, e caso houver, se este ganho justifique o custo computacional adicional de processamento.

1.2.2 Objetivos Específicos

- Reproduzir as métricas obtidas por um algoritmo de alto desempenho em uma categoria que contenha cenários típicos observados por câmeras de segurança. Esta reprodução é necessária para que se tenha uma base que o algoritmo possui o mesmo desempenho encontrado na literatura.
- Aplicar o algoritmo escolhido num conjunto de dados de resoluções variadas gerado a partir de um vídeo gravado em resolução 4k (3840×2160).
- Medir o desempenho do algoritmo e o custo computacional do algoritmo para cada resolução.

1.3 ORGANIZAÇÃO DO TEXTO

Este trabalho de conclusão de curso está organizado da seguinte forma. No capítulo 2 são apresentados os principais conceitos sobre extração de elementos de interesse, seguidos de uma discussão sobre os métodos de extração de elementos de interesse que representam o estado da arte, e por fim uma explanação do algoritmo escolhido e o dataset utilizado. O capítulo 3

[2] Changedetection.net

contém os recursos utilizados para a realização deste estudo, se tratando de software e hardware, bem como qual foi o procedimento para mensurar as métricas e os recursos computacionais gastos. No capítulo 4 é dedicado a uma análise dos resultados obtidos. Por fim o capítulo 5 é apresentada a conclusão do trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, serão apresentados os principais conceitos sobre visão computacional, o processo de extração de elementos de interesse de vídeos, uma visão detalhada sobre o algoritmo SubSENSE além dos trabalhos relacionados com a presente proposta.

2.1 VISÃO COMPUTACIONAL

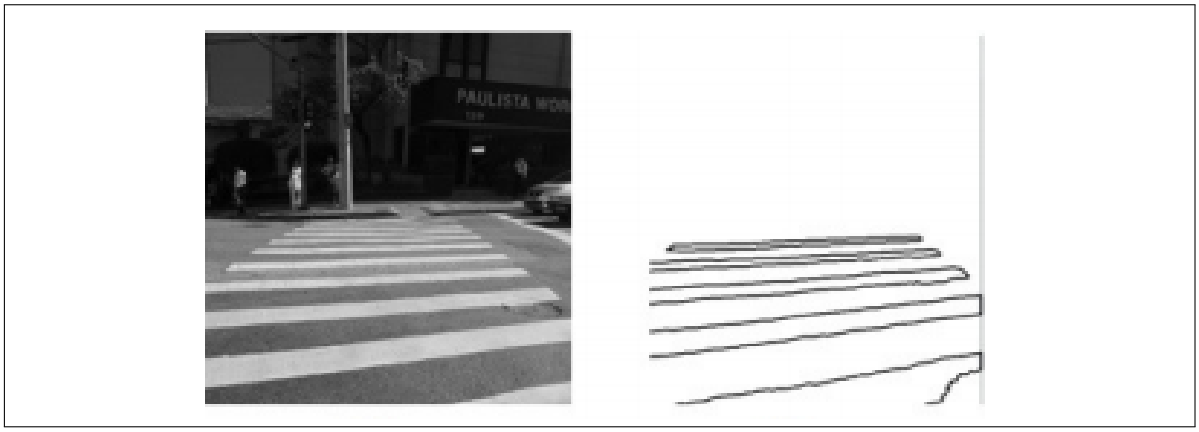
Segundo **??**), a visão computacional está diretamente relacionada com o processamento de imagens, porém diferem em alguns aspectos. A correlação entre ambas as áreas, segundo o autor, pode ser definida considerando três níveis no espectro que vai do processamento de imagens até a visão computacional.

Os processos de baixo-nível envolvem operações primitivas, tais como a redução de ruído ou melhoria no contraste de uma imagem. O segundo nível, ao qual pertencem os processos de nível-médio, são operações do tipo segmentação (particionamento da imagem em regiões) ou classificação (reconhecimento dos objetos na imagem). Os processos que pertencem ao alto nível estão relacionados com as tarefas de cognição que estão associadas com a visão humana (**??**).

Em outras palavras, o processamento de imagens é um processo no qual a entrada do sistema é uma imagem e a saída é um conjunto de valores numéricos, que podem ou não compor uma outra imagem. A visão computacional, procura emular a visão humana, também possui como entrada uma imagem, no entanto, a saída é uma interpretação da imagem como um todo, ou parcialmente (**??**).

Na Figura 1 pode ser visualizada um exemplo de aplicação da visão computacional em que o sistema faz o reconhecimento de faixas de pedestres em uma imagem.

Figura 1 – Visão computacional em um sistema de reconhecimento de faixas de pedestres.



Fonte: (??)

2.1.1 Extração de Elementos de Interesse

A segmentação (extração de elementos de interesse em imagens ou vídeos) é uma área de estudo da indústria de vídeo desde o começo do século 20 (**??**). Muitas das aplicações nessa área geralmente extraem das cenas os elementos de interesse (geralmente pessoas) de seu fundo original, com o objetivo de criar um novo cenário em que o elemento extraído é inserido (**??????**).

Os métodos mais tradicionais assumem que o vídeo foi capturado em um ambiente controlado, com um fundo de uma única cor e iluminação disposta de tal forma que a cor desse fundo seja mantida uniforme (Figura 2). Desse modo, a segmentação pode ser feita em tempo real e com baixa taxa de erro (**??????**).

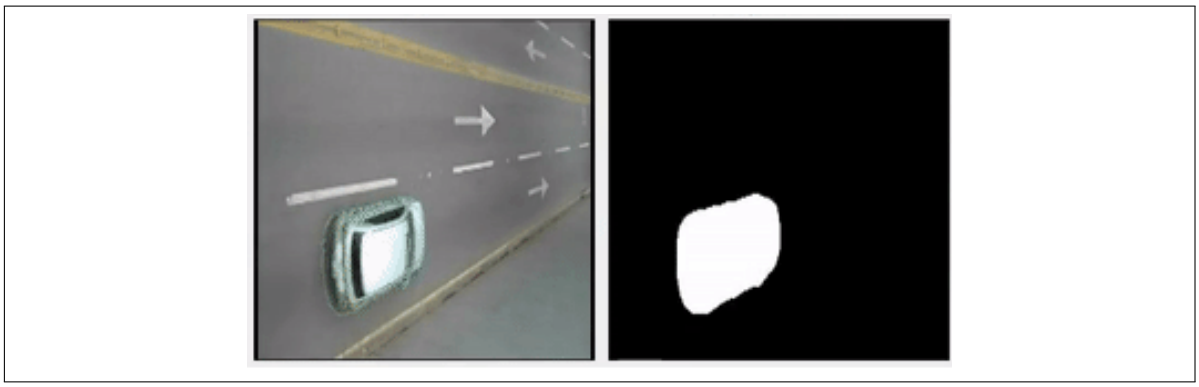
Figura 2 – Exemplo de extração do elemenbto de interesse e adição de um fundo artificial em uma imagem com fundo uniforme.

Fonte: (??)

Desde a década de 80, a tecnologia necessária e os algoritmos de extração de elementos de interesse têm sido aprimorados de tal modo que novos métodos são capazes de extrair elementos não apenas em tempo real mas também de imagens naturais (sem uma cor uniforme de fundo) foram desenvolvidos (**??**). As novas aplicações que passaram a utilizar esses métodos são várias, como as videoconferências (ou *videochats*) em que os quadros são processados e o fundo original é substituído antes de a imagem ser enviada a um usuário remoto (**????????**).

A extração de elementos de interesse de quadros de vídeo também é uma tarefa necessária em diversas aplicações voltadas para segurança por vídeo (**??**). Exemplos de aplicações nessa área são a detecção de quedas de pessoas, a contagem de veículos em rodovias, a detecção de acidentes etc. A segmentação do quadro de vídeo é a primeira tarefa a ser executada em sistemas desse tipo. A Figura 3 mostra uma cena típica de uma aplicação de segurança por vídeo e a máscara gerada (contendo o elemento de interesse) pelo algoritmo de segmentação.

Figura 3 – Cena de uma aplicação de segurança por vídeo voltada para contar veículos em rodovias e a máscara gerada pelo algoritmo de segmentação.



Fonte: (??)

2.2 MÉTODOS DE EXTRAÇÃO DE ELEMENTOS DE INTERESSE

A separação entre o fundo da imagem e o elemento de interesse é um passo necessário em muitas aplicações na área da visão computacional. Muitos novos algoritmos, baseados em diferentes abordagens, são apresentados e podem ser encontrados na literatura. Segundo **??**), a maioria das soluções compartilham um mesmo esquema básico:

- Inicialização do fundo: neste passo, o algoritmo tem como objetivo criar um modelo do fundo com base em um número de quadros da imagem. Este modelo pode ser criado de vários métodos (estatístico, fuzzy, inspirado em redes neurais).
- Detecção do fundo: nos próximos quadros, o algoritmo irá realizar uma comparação entre o modelo criado e a imagem atual. A subtração resultará no reconhecimento do fundo da imagem.
- Manutenção do modelo do fundo: um processo contínuo que visa atualizar o modelo criado na fase de inicialização, analisando as imagens ao decorrer do tempo e identificando como o fundo da imagem está mudando.

No trabalho de **??**), os algoritmos que representam o estado-da-arte são classificados em quatro grupos: básicos, estatísticos, fuzzy, e de redes neurais. Os métodos básicos são aqueles que não utilizam modelos estatísticos, fuzzy ou redes neurais. A extração de elementos de interesse consiste em criar um modelo para o fundo da imagem. Isso pode ser feito, por exemplo, ao se definir uma imagem do fundo sem nenhum objeto em movimento. A partir desse modelo, é feita uma análise do quadro atual, com base na comparação com o quadro a imagem de fundo. Essa abordagem é chamada de Diferença Estática de Quadros (DEQ). A Figura 6 mostra exemplos de um quadro atual e do fundo estático capturado no início do processo (**??**).

Figura 4 – Quadro de vídeo sendo processado e imagem capturada do fundo (sem o elemento de interesse).

Fonte: (??)

2.3 MÉTODO ADOTADO: SUBSENSE

Após uma análise dos métodos de segmentação de elementos de interesse da presentes na Background Substraction Library (BGSLIBRARY) juntamente com os resultados do site Changedetection (CDnet)[1] foi escolhido o método “Self-Balanced SENsitivity SEgmenter” (SubSENSE) (**??**). Nesta seção serão abordados as características principais do SubSENSE que o fizeram ser escolhido como algoritmo de estudo e do conjunto de dados utilizado: o *dataset* CDnet 2012.

2.3.1 Método SuBSENSE: Teoria

O método SubSENSE é uma adaptação e integração de recursos do Local Binary Similarity Pattern (LBSP) (em portugês “Padrões locais binários de similaridade"), um método que utiliza a distância de Hamming para detecção sutis no cenário em questão, proposto por **??**) e demonstrado mais eficaz que detecção de mudança baseada comparações de cores. Este método é utilizado para criação de um modelo de fundo não paramétrico que por sua vez é ajustado automaticamente utilizando repetições baseadas em *feedback* a nível de pixel.

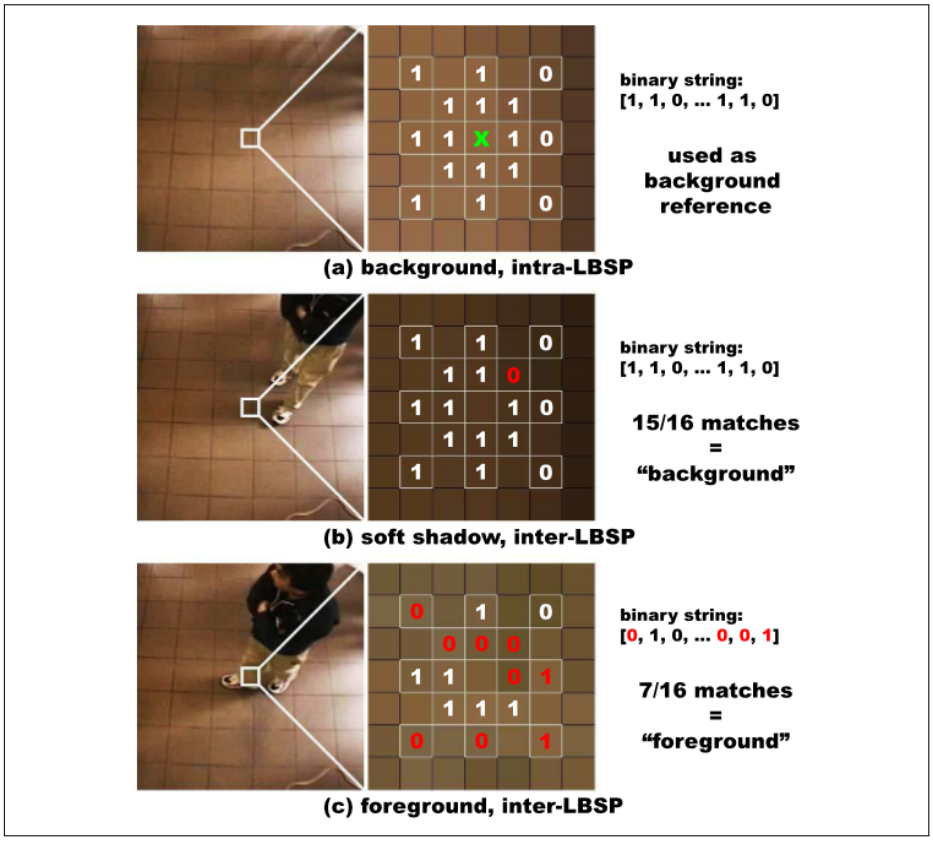
O processo de criação deste modelo é representado na figura 5, que é explicada pelos seguintes passos:

- No quadro (a) é definida uma string binária que é utilizada como referência do que é o fundo da imagem.
- No quadro seguinte é feita a comparação da mesma região, e como o número de acertos ficou acima de um número pré-definido no momento da criação do modelo, o local ainda é considerado fundo.

[1] changedetection.net

- No quadro da imagem (c) o número de acertos ficou abaixo do número pré-definido, portanto sendo classificado como objeto de interesse.

Figura 5 – Processo de classificação LBSP.



Fonte: (??)

Já na equação 1, temos i_x que é a “referência central” e corresponde à intensidade do pixel localizado em x, i_p corresponde à intensidade do vizinho *p*th no padrão pré-definido (uma matriz $5x5$, por exemplo), e T_r é o *threshold* (em português “limiar”) interno de similaridade, fixado entre 0 e 1. Esta equação tem como resultado uma distância entre os *pixels* i_x e i_p e seu resultado é utilizado na equação 3.

$$d(i_p, i_x) = \begin{cases} 1 & se \quad |i_p - i_x| \leq T_r \cdot i_x \\ 0 & caso\ contrário \end{cases} \tag{1}$$

A equação 2 define B, que representa todos os *pixels* do *background* (o valor B(x) representa um pixel da imagem) $B_1(x)$, $B_2(x)$,..., representam amostras do background em um determinado tempo t. Estas amostras são comparadas com seus respectivos quadros observados no tempo t em que elas foram coletadas, denotado por $I_t(x)$, para realizar a classificação do pixel na coordenada x como objeto de interesse (1) ou fundo (0), conforme mostrado na equação

3.

$$B(x) = \{B_1, B_2, \ldots, B_N(x)\} \tag{2}$$

Na esquação 3, é calculada a distância de cada uma das n amostras com o pixel observado no tempo t (é proposto que sejam usadas entre 35 e 50 amostras). Se todas as comparações forem verdadeiras, o pixel é definido como 1. Em outras palavras, o pixel x da imagem atual é similar ao pixel do modelo do fundo que o sistema mantém.

$$S_t(x) = \begin{cases} 1 \;\; se \quad \#\{dist\,(I_t\,(x)\,, B_n(x)) < \; R, \forall n\}\, \#_{min} \\ 0 \qquad\qquad\qquad\qquad caso\, contrário \end{cases} \tag{3}$$

Figura 6 – Ranking de acordo com o F-Measure da categoria baseline do conjunto de dados 2012.

Method	Average ranking	Average Re	Average Sp	Average FPR	Average FNR	Average PWC	Average F-Measure
SuBSENSE [38]	4.43	0.9520	0.9982	0.0018	0.0480	0.3574	0.9503
CDet [36]	8.14	0.9704	0.9974	0.0026	0.0296	0.3589	0.9458
PAWCS [37]	8.43	0.9408	0.9980	0.0020	0.0592	0.4491	0.9397
STBM [46]	12.00	0.9524	0.9973	0.0027	0.0476	0.3840	0.9345
SC-SOBS [16]	8.14	0.9327	0.9980	0.0020	0.0673	0.3747	0.9333
Spectral-360 [26]	15.29	0.9615	0.9968	0.0032	0.0385	0.4263	0.9330
SOBS_CF [34]	11.00	0.9347	0.9978	0.0022	0.0653	0.3912	0.9299
PSP-MRF [9]	12.14	0.9319	0.9978	0.0022	0.0681	0.4127	0.9289
Multimode Background Subtraction Version 0 (MBS V0) [40]	12.00	0.9158	0.9979	0.0021	0.0842	0.4361	0.9287
Multimode Background Subtraction(MBS) [41]	12.00	0.9158	0.9979	0.0021	0.0842	0.4361	0.9287

Fonte: changedetection.net

2.3.2 Dataset escolhido: CDnet dataset 2012

Em 2012 no Computer Vision and Pattern Recognition Workshops (CPVRW) on Change Detection (em português “Seminário de Visão Computacional e Reconhecimento de Padrões em detecção de mudanças”) foi introduzido um método de análise de performance (*benchmark*) que diferentemente de seus predecessores, proveu uma grande variedade de cenários de segmentação ocorrendo em condições realistas com um conjunto de dados *groundtruth* preciso (**??**).

Este conjunto de dado possui 6 categorias, sendo elas:

- *Baseline*: esta categoria contém uma mistura de vídeos das outras categorias do conjunto de dados, com exceção da categoria “thermal"que possui vídeos capturados com câmeras

com tecnologia infravermelho.

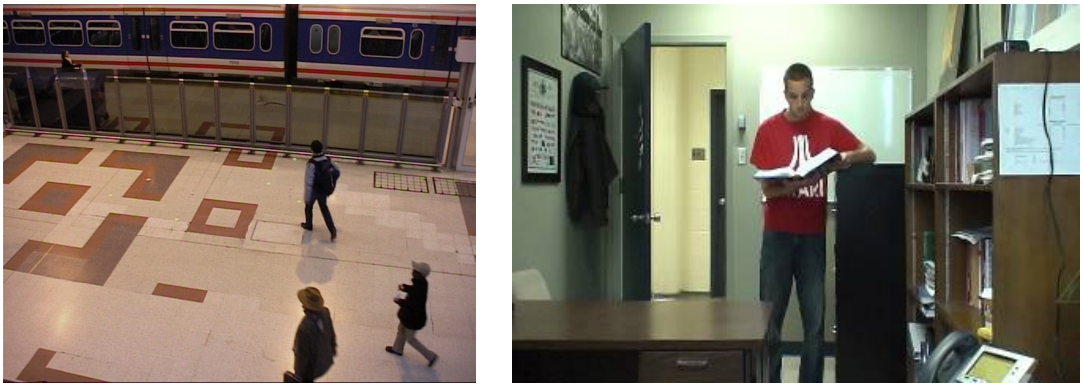
- *Dynamic Background*: categoria com vídeos em que o fundo possui constante movimento. Exemplo: carros passando perto de fontes de água, barcos no mar, etc.
- *Camera Jitter*: categoria com imagens capturadas em ambientes internos e externos com câmeras instáveis.
- *Intermittent Object Motion*: categoria com vídeos em cenários conhecidos por causar o efeito de *ghosting*, isto é, vídeos em que o objeto de interesse está em movimento, para subitamente e volta a entrar em movimento.
- *Shadows*: categoria com imagens capturadas em ambientes fechados e abertos nos quais há a presença de sombras.
- *Thermal*: categoria que possui vídeos capturados com câmeras com tecnologia infravermelho.

O conjunto de dados completo possui 31 vídeos capturados por câmeras e mais de 70.000 *frames*. Após uma análise, optou-se por utilizar a categoria *baseline* pelo motivo da mesma ser um compilado dos desafios encontrados nas outras. Esta categoria contém 4 vídeos, totalizando 6049 *frames*. Exemplos de cada vídeo podem ser observados nas imagens 7 e 8.

Figura 7 – Imagens dos vídeos *Highway* e *Pedestrians*.

Fonte: changedetection.net

Figura 8 – Imagens dos vídeos *PETS2006* e *office*.



Fonte: changedetection.net

2.4 TRABALHOS RELACIONADOS

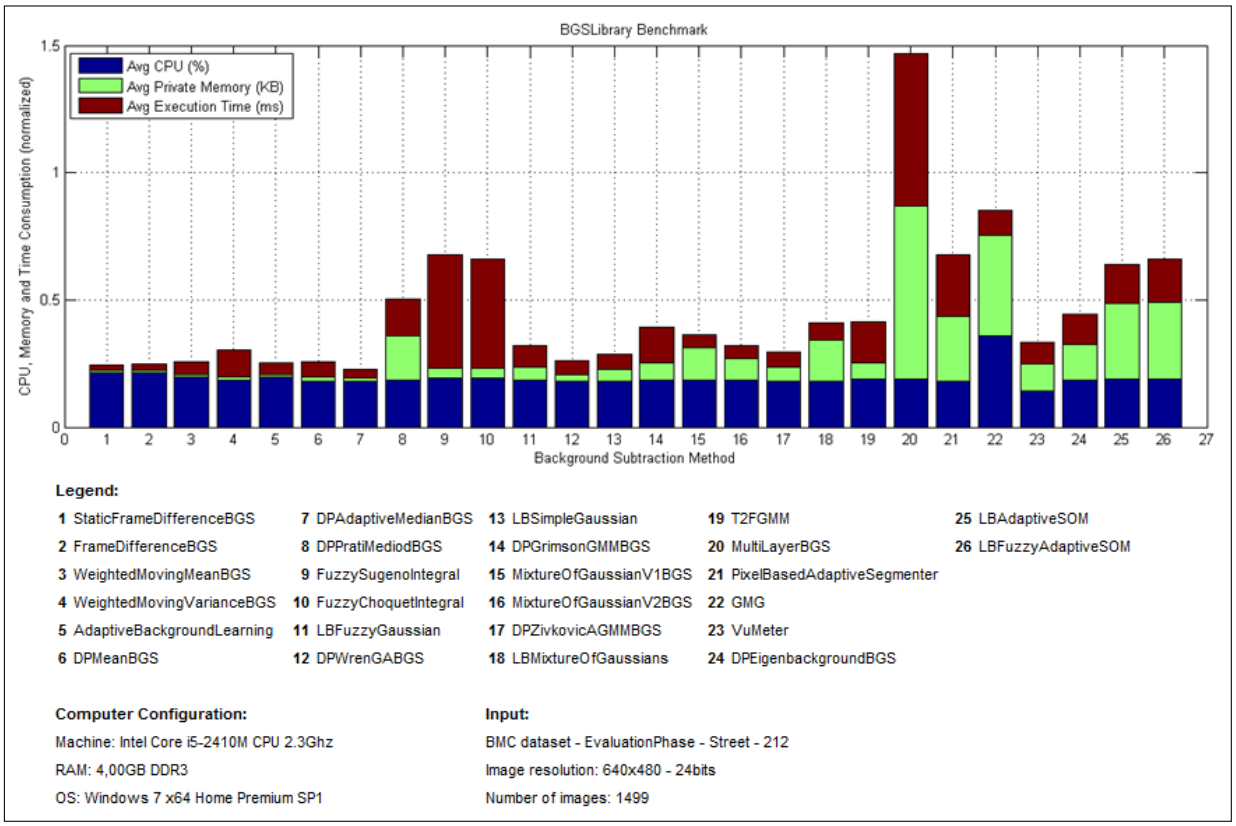
Desempenho de algoritmos de extração de elementos de interesse em imagens de alta resolução não é algo muito abordado pelos trabalhos presentes na literatura, uma vez que a disponibilidade de câmeras capazes de capturar imagens com tais resoluções, é algo recente (**??**).

No trabalho de **??**), imagens de alta resolução foram utilizadas para avaliar o desempenho do método proposto, que tem como objetivo a extração de malhas rodoviárias localizadas dentro de perímetros urbanos. Métodos desse tipo são utilizados em aplicações como os sistemas de navegação. Uma vez que oo grande desafio desses algoritmos é diferenciar os limites de uma via urbana do seu plano de fundo, imagens com resoluções maiores são mais recomendadas, pois nelas os limites são mais perceptíveis.

Entre os trabalhos focados em segmentação de elemento de interesse, pode-se destacar o apresentado por **??**) em que a segmentação se baseia em características temporais, além das informações espaciais de cores, para detectar alterações em quadros consecutivos que permitam identicar o elemento de interesse. O algoritmo tem como principal contribuição a extração correta do elemento de interesse mesmo quando este mostra-se camuflado com objetos do fundo. Ajustes dinâmicos de parâmetros são realizados quadro a quadro, considerando o contexto em que se encontram e operações de baixo nível são aplicadas para eliminar ruídos.

??) além da compilação em um único *framework* de vários algoritmos encontrados na literatura, contribuiu com um teste de performance dos algoritmos implementados. A figura 9 mostra os valores de CPU, memória consumida e tempo de execução de cada um dos 26 algoritmos originalmente implementados.

Figura 9 – Teste de performance dos algoritmos encontrados na BGSLIBRARY.



Fonte: (??)

3 MATERIAIS E MÉTODOS

Neste capítulo serão listadas as ferramentas utilizadas e os métodos adotados.

3.1 FERRAMENTAS

Para a execução destre trabalho, os seguintes itens foram utilizados:

- IDE com suporte a C++, CodeBlocks versão 16.01.
- MATLAB 2013.
- Ferramenta Monitor de Desempenho do Windows 10.
- Conjunto de dados CDnet 2012.
- Câmera para captura de vídeo 4k (3840 x 2160 pixels).

A IDE CodeBlocks foi utilizada para execução do algoritmo SubSENSE obtida na BGSLIBRARY

Foi utilizado um computador com um processador Intel i5 de 2.6 GHz, 8 GB de memória RAM, 1 TB de armazenamento e sistema operacional Windows 10.

3.2 ALGORITMO UTILIZADO: SUBSENSE

A implementação utilizada do método de extração de elementos de interesse SubSENSE está disponível online, como parte do *framework* BGSLIBRARY[1]. Algumas alterações no código C++ foram feitas, para facilitar o *input* de tipos diferentes de arquivos, conforme o necessário. Todos os códigos utilizados neste trabalho, estão disponíveis no GitHub[2].

Para um teste inicial, de forma a garantir que o algoritmo tenha a mesma acurácia obtida na literatura, foi o mesmo foi executado no *dataset* CDnet 2012. Conforme explicado na seção 2, foram utilizadas as imagens da categoria baseline, obtendo-se máscaras que representam as regiões da imagem em que o objeto de interesse se encontram.

No site changedetection.net está disponível um *script* de MATLAB para calcular as métricas que avaliam o desempenho dos algoritmos, uma parte do algoritmo está mostrada no algoritmo 1.

[1] https://github.com/andrewssobral/bgslibrary/
[2] https://github.com/mercante/subsense4k

Algoritmo 1 – Trecho de código que calcula as métricas

```
function [TP FP FN TN SE stats] =
  ↪ confusionMatrixToVar(confusionMatrix)
    TP = confusionMatrix(1);
    FP = confusionMatrix(2);
    FN = confusionMatrix(3);
    TN = confusionMatrix(4);
    SE = confusionMatrix(5);

    recall = TP / (TP + FN);
    specficity = TN / (TN + FP);
    FPR = FP / (FP + TN);
    FNR = FN / (TP + FN);
    PBC = 100.0 * (FN + FP) / (TP + FP + FN + TN);
    precision = TP / (TP + FP);
    FMeasure = 2.0 * (recall * precision) / (recall + precision);

    stats = [recall specficity FPR FNR PBC precision FMeasure];
```

As métricas são calculadas a partir das seguintes observações obtidas da comparação da máscara gerada com o *groundtruth* presente no *dataset*:

- TP: True Positive (positivo verdadeiro), pixel presente no objeto de interesse e classificado como tal.
- FP: False Positive (positivo falso), pixel não presente no objeto de interesse e classificado como tal.
- FN: False Negative (falso negativo), pixel presente no objeto de interesse e não classificado como tal.
- TN: True Negative (negativo verdadeiro), pixel não presente no objeto de interesse e classificado como tal.

As métricas são definidas pelas seguintes equações utilizando as observações anteriormente listadas:

- *Recall*:

$$Re = \frac{TP}{TP+FN} \tag{1}$$

- *Specificity*:

$$Sp = \frac{TN}{TN+FP} \tag{2}$$

- False Positive Rate (FPR):

$$FPR = \frac{FP}{FP+TN} \tag{3}$$

- False Negative Rate (FNR):

$$FNR = \frac{FN}{TP+FN} \tag{4}$$

- Porcentage of Wrong Classifications (PWC):

$$PWC = 100 * \frac{FN+FP}{TP+FN+FP+TN} \tag{5}$$

- Precision:

$$Precision = \frac{TP}{TP+FP} \tag{6}$$

- F-Measure:

$$FMeasure = \frac{2 * Precision * Recall}{Precision\ +\ Recall} \tag{7}$$

O quadro 1 apresenta os resultados obtidos utilizando algoritmo SubSENSE e o *script* disponível no site changedetection.net.

Quadro 1 – Métricas do dataset CDnet 2012 categoria Baseline

Dataset	Recall	Specificity	FPR	FNR	PWC	Precision	FMeasure
Baseline	0.9529	0.9981	0.0018	0.0470	0.3586	0.9484	0.9503

Fonte: Autoria própria

Como podemos ver na figura 10, retirada do próprio site changedetection.net os valores obtidos correspondem aos encontrados na literatura.

Figura 10 – Resultado da categoria baseline listados do maior F-Measure para o menor.

Method	Average ranking	Average Re	Average Sp	Average FPR	Average FNR	Average PWC	Average F-Measure
SuBSENSE [38]	4.43	0.9520	0.9982	0.0018	0.0480	0.3574	0.9503
CDet [36]	8.14	0.9704	0.9974	0.0026	0.0296	0.3589	0.9458
PAWCS [37]	8.43	0.9408	0.9980	0.0020	0.0592	0.4491	0.9397
STBM [46]	12.00	0.9524	0.9973	0.0027	0.0476	0.3840	0.9345
SC-SOBS [16]	8.14	0.9327	0.9980	0.0020	0.0673	0.3747	0.9333
Spectral-360 [26]	15.29	0.9615	0.9968	0.0032	0.0385	0.4263	0.9330
SOBS_CF [34]	11.00	0.9347	0.9978	0.0022	0.0653	0.3912	0.9299
PSP-MRF [9]	12.14	0.9319	0.9978	0.0022	0.0681	0.4127	0.9289
Multimode Background Subtraction Version 0 (MBS V0) [40]	12.00	0.9158	0.9979	0.0021	0.0842	0.4361	0.9287
Multimode Background Subtraction(MBS) [41]	12.00	0.9158	0.9979	0.0021	0.0842	0.4361	0.9287

Fonte: changedetection.net

3.3 MÉTODO UTILIZADO PARA O TESTE DE PERFORMANCE EM IMAGENS DE ALTA RESOLUÇÃO

Para o desenvolvimento deste trabalho, foi gravado um vídeo de aproximadamente 17 segundos em resolução 4k (3840 x 2160). Este vídeo foi gravado utilizando um celular chamado "Moto Z Play". A partir deste vídeo, foi criado um *dataset* chamado de "Cornélio4k", que é composto pelos *frames* do vídeo em sua resolução original e outras 9 obtidas através da função de redimensionamento textitimresize que utiliza interpolação de proximidade. Todas resoluções foram geradas de modo a se manter o aspecto 16:9. Também parte do *dataset*, está presente um *groundtruth* utilizado para a avaliação da segmentação. As resoluções disponíveis deste *dataset* estão mostradas no quadro 2.

Quadro 2 – Quadro das resoluções dos vídeos utilizados no teste de performance

Vídeo	Resolução
Cornelio384	384 x 216
Cornelio768	768 x 432
Cornelio1152	1152 x 648
Cornelio1536	1536 x 864
Cornelio1920	1920 x 1080
Cornelio2304	2304 x 1296
Cornelio2688	2688 x 1512
Cornelio3072	3072 x 1728
Cornelio3456	3456 x 1944
Cornelio3840	3840 x 2160

Fonte: Autoria própria

Durante a execução do algoritmo, com o auxílio da ferramenta “Monitor de Performance” do Windows, foram monitorados as seguintes variáveis:

- “Tempo de execução”: se trata do tempo total transcorrido, em segundos, desde o início da execução do processo.
- “Memória RAM”: é o tamanho em *bytes*, da memória alocada pelo processo que não pode ser compartilhada com outros processos.
- “% tempo do processador”: é a porcentagem de tempo decorrido em que todos os *threads* de processo usaram o processador para executar instruções. Uma instrução é a unidade básica de execução em um computador, um *thread* é o objeto que executa instruções e um processo é o objeto criado quando um programa é executado. O código executado para lidar com algumas interrupções de hardware e condições de desvio é incluído nessa contagem.

Os resultados obtidos são discutidos no capítulo 4.

Por fim, utilizando o *script* do MATLAB de obtenção de métricas anteriormente mencionado, foi calculada a acurácia da segmentação em todas as resoluções para poder ser feita uma análise de como a resolução está relacionada com a eficácia da segmentação, e se este aumento de acurácia, caso exista, justifica o custo computacional adicional.

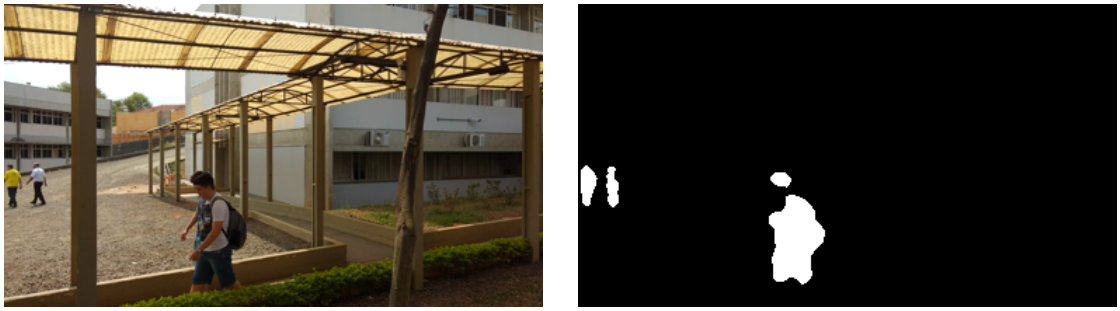
4 ANÁLISE DOS RESULTADOS

A segmentação das imagens do “dataset” Cornelio4k pode ser observada nas figuras 11, 12 e 13. Nas imagens da direita temos a máscara gerada pelo algoritmo SubSENSE e à esquerda a imagem de entrada. Estas máscaras foram utilizadas para medir a acurácia da segmentação comparando-as com seu arquivo *groundtruth* correspondente, como pode ser visto na figura 14.

Figura 11 – Resultado da segmentação utilizando o algoritmo SubSENSE - 1.

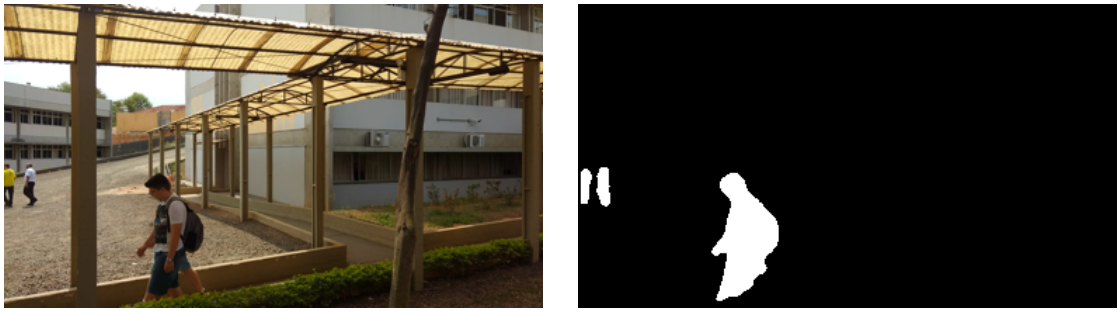
Fonte: Autoria própria

Figura 12 – Resultado da segmentação utilizando o algoritmo SubSENSE - 2.



Fonte: Autoria própria

Figura 13 – Resultado da segmentação utilizando o algoritmo SubSENSE - 3.



Fonte: Autoria própria

Figura 14 – Resultado da segmentação e seu *groundtruth* correspondente.

Fonte: Autoria própria

Com o *script* do MATLAB para obtenção de métricas de acurácia encontrado no site changedetection.net, foram obtidas as métricas do quadro 3. Este quadro é resultado de uma análise do conjunto como um todo, mais à frente serão apresentados os resultados de cada resolução individual. É relevante destacar o F-Measure obtido de 0,7938, que é um resultado bastante satisfatório quando levado em conta que o algoritmo SubSENSE, utilizado para a extração dos elementos de interesse, foi aplicado com os parâmetros sugeridos por **??**).

Quadro 3 – Métricas do dataset Cornelio4k

Dataset	Recall	Specificity	FPR	FNR	PWC	Precision	FMeasure
Cornelio4k	0.7864	0.9903	0.0097	0.2136	1.9394	0.8020	0.7938

Fonte: Autoria própria

Enquanto o algoritmo era executado, foram obtidos os valores de Tempo de Execução, Memória RAM e CPU (% processor time), discutidos na seção anterior deste trabalho. No quadro 4 temos os valores totais medidos pelo “Monitor de Desempenho” do Windows.

Para uma melhor visualização, os gráficos ilustrados nas figuras 15, 16, 18 e 17 possuem os valores obtidos. Como podemos observar, embora a memória e o tempo de execução cresçam

Quadro 4 – Valores do teste de performance

Input	Tempo de Execução (s)	Memória RAM (MB)	CPU (%processortime)
Cornelio384	57,030	68,728	87,654
Cornelio768	832,867	209,632	95,997
Cornelio1152	1918,954	443,470	97,851
Cornelio1536	2989,303	769,987	98,966
Cornelio1920	6415,051	1193,885	97,889
Cornelio2304	7799,348	1707,796	98,003
Cornelio2688	9065,279	2316,898	99,232
Cornelio3072	11083,866	3022,250	99,648
Cornelio3456	16549,882	3729,219	99,170
Cornelio3840	17748,630	4598,337	99,430

Fonte: Autoria Própria

gradativamente conforme a resolução aumenta, o CPU($\%$ processor time) mantém-se constante a partir da segunda resolução. Com isto podemos tirar uma conclusão de que uma maior quantidade de memória é essencial quando se trabalha com imagens de alta resolução, enquanto o processador utilizado neste trabalho se mostrou suficiente nos experimentos utilizados.

Quadro 5 – Valores em porcentagem da razão de uma resolução em relação à sua anterior

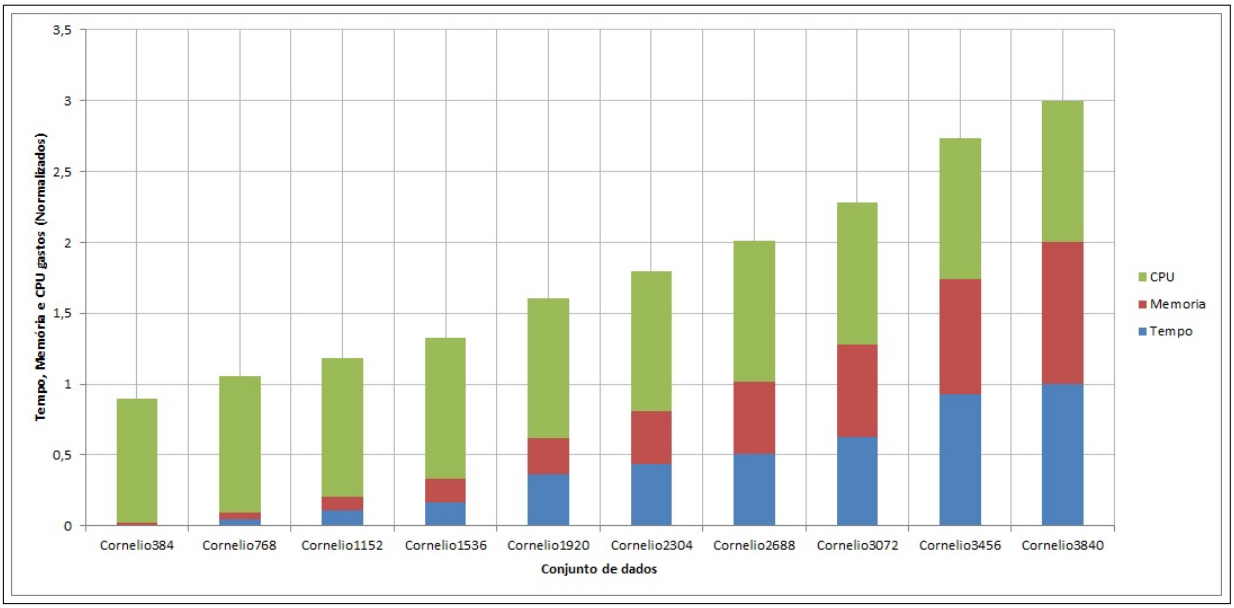
Input	Tempo de Execução (s)	Memória RAM (MB)	CPU (%processortime)
R1	1360%	205%	10%
R2	130%	112%	2%
R3	56%	74%	1%
R4	115%	55%	-1%
R5	22%	43%	0%
R6	16%	36%	1%
R7	22%	30%	0%
R8	49%	23%	0%
R9	7%	23%	0%
Total	1778%	601%	13%

Fonte: Autoria Própria

A partir dos dados do quadro 4, foi gerado outro o 5. Este contem as porcentagens de variação entre cada uma das resoluções. Por exemplo, a razão do *input* Cornelio768 pela Cornelio384 é chamada de R1, e assim por diante. Novamente, vemos que o tempo de execução e a memória são as duas variáveis que sofrem maior variação entre as resoluções.

Os quadros 6 e 7 mostram as métricas de acurácia obtidos, em valores absolutos e a razão de uma resolução para outra, respectivamente. No quadro 7, a linha em vermelho destaca onde houve o maior aumento na acurácia, que foi do input Cornelio1536 em relação

Figura 15 – Gráfico dos resultados obtidos da performance total.



Fonte: Autoria própria

ao Cornelio1152. Se observarmos os outros gráficos de consumo de recursos computacionais anteriormente discutidos, podemos chegar à conclusão que à partir desta resolução os resultados de acurácia não justificam o investimento adicional em recursos computacionais para processar imagens obtidas em resolução superior.

Quadro 6 – Métricas por resolução

Input	Recall	Specificity	FPR	PWC	Precision	FMeasure
Cornelio384	0,6680	0,9884	0,0116	2,6853	0,7422	0,7032
Cornelio768	0,7357	0,9878	0,0122	2,4034	0,7486	0,7421
Cornelio1152	0,7551	0,9836	0,0164	2,7280	0,6972	0,7250
Cornelio1536	0,8185	0,9912	0,0088	1,7043	0,8228	0,8206
Cornelio1920	0,8179	0,9913	0,0087	1,6915	0,8252	0,8216
Cornelio2304	0,8175	0,9915	0,0085	1,6733	0,8275	0,8225
Cornelio2688	0,8141	0,9919	0,0081	1,6523	0,8347	0,8243
Cornelio3072	0,8129	0,9922	0,0078	1,6341	0,8388	0,8256
Cornelio3456	0,8107	0,9926	0,0074	1,6092	0,8449	0,8275
Cornelio3840	0,8131	0,9923	0,0077	1,6129	0,8389	0,8258

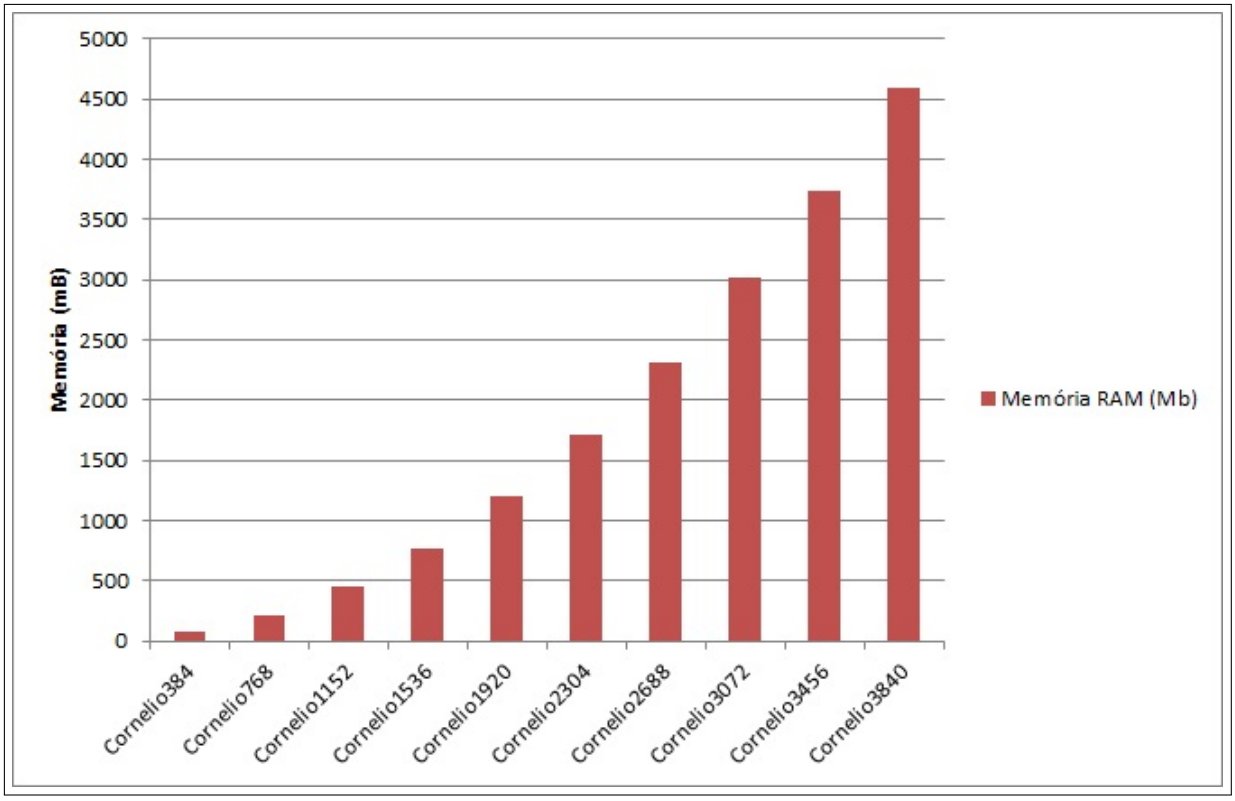
Fonte: Autoria própria

Quadro 7 – Razão das métricas por resolução

Input	Recall	Specificity	FPR	PWC	Precision	Fmeasure
R1	10,1230%	-0,0590%	5,0243%	-10,4953%	0,8664%	5,5352%
R2	2,6383%	-0,4267%	34,6014%	13,5026%	-6,8736%	-2,3072%
R3	8,4018%	0,7706%	-46,2258%	-37,5244%	18,0159%	13,1963%
R4	-0,0676%	0,0158%	-1,7795%	-0,7517%	0,2933%	0,1121%
R5	-0,0483%	0,0179%	-2,0481%	-1,0747%	0,2815%	0,1156%
R6	-0,4177%	0,0427%	-4,9866%	-1,2559%	0,8617%	0,2140%
R7	-0,1558%	0,0255%	-3,1391%	-1,1052%	0,4925%	0,1633%
R8	-0,2607%	0,0372%	-4,7226%	-1,5190%	0,7305%	0,2247%
R9	0,2908%	-0,0267%	3,5590%	0,2251%	-0,7074%	-0,2005%
Total	20,5037%	0,3973%	-19,7169%	-39,9985%	13,9607%	17,0534%

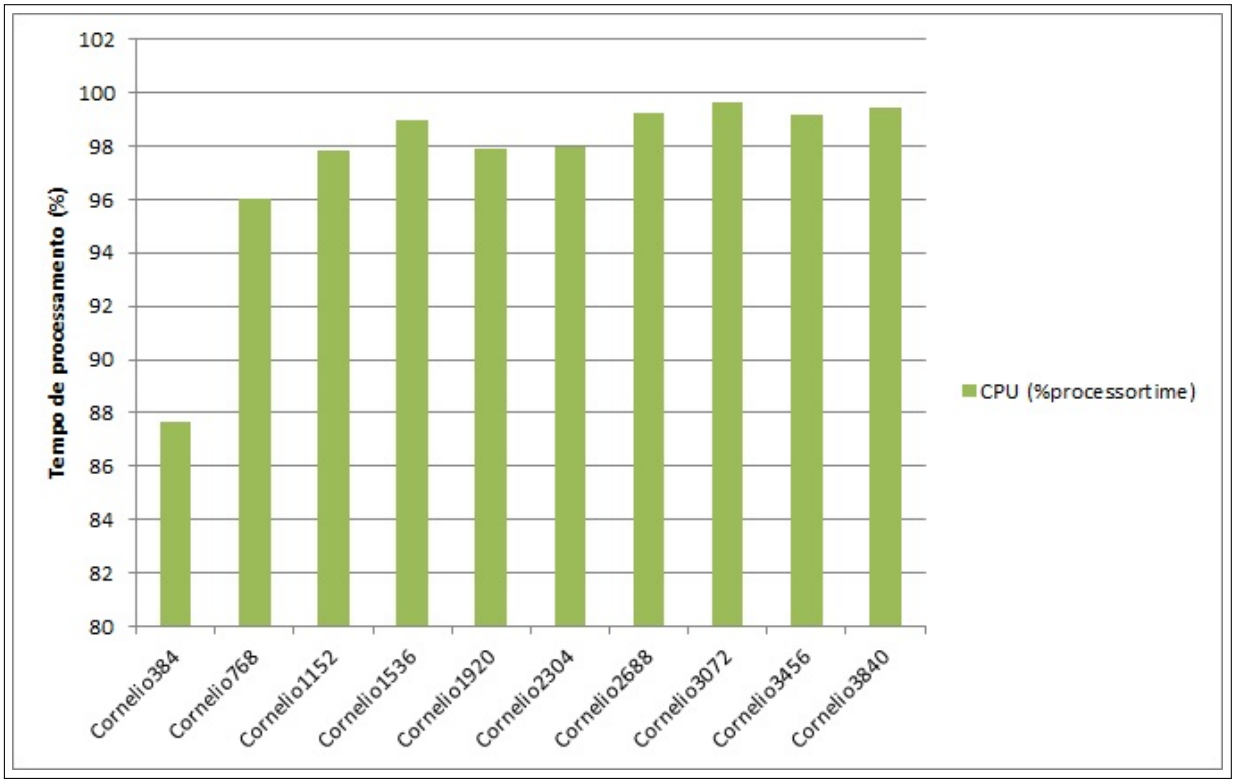
Fonte: Autoria Própria

Figura 16 – Gráfico dos resultados obtidos da performance da memória.



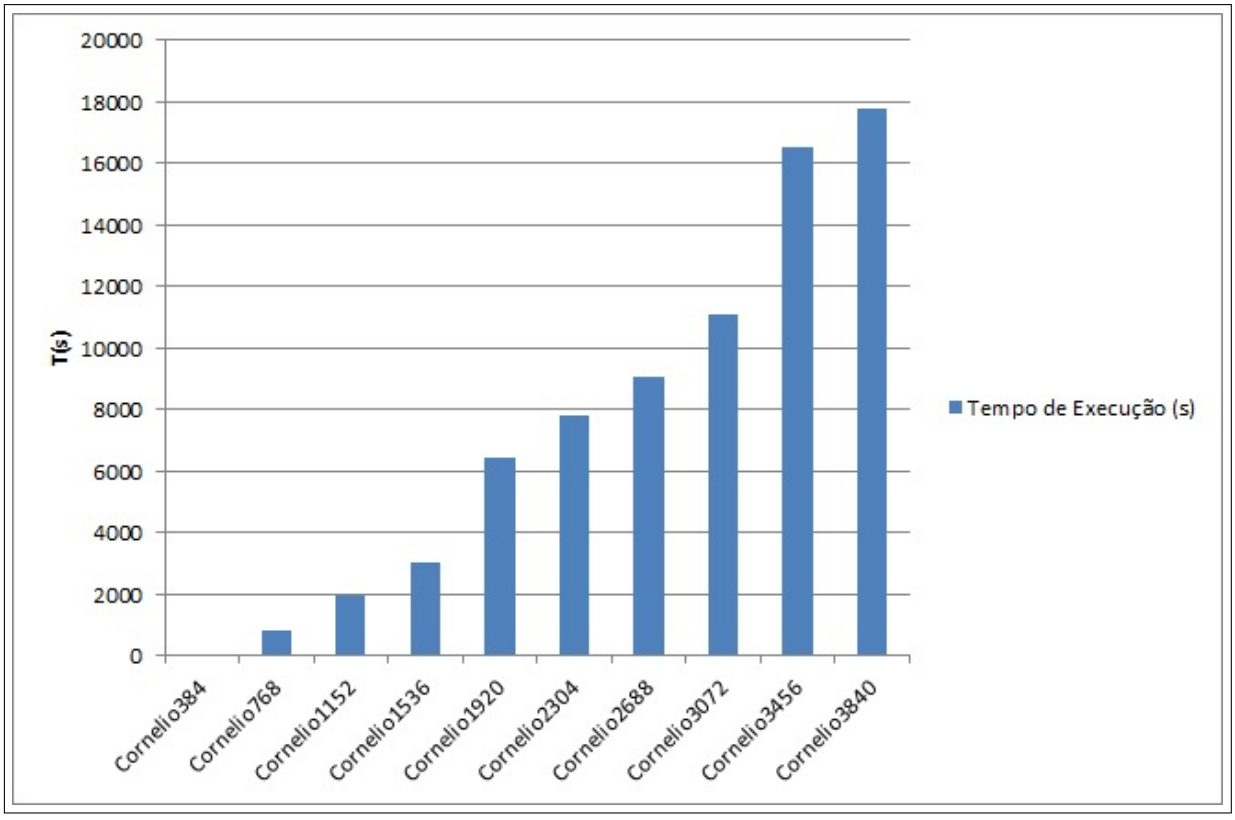
Fonte: Autoria própria

Figura 17 – Gráfico dos resultados obtidos da performance do cpu.



Fonte: Autoria própria

Figura 18 – Gráfico dos resultados obtidos da performance em relação ao tempo.



Fonte: Autoria própria

5 CONCLUSÃO

O trabalho proposto teve como objetivo realizar um estudo do desempenho de um algoritmo de segmentação de vídeos em imagens de alta resolução (4k) e correlacionar os resultados obtidos com o custo computacional necessário para o processamento destas imagens. Através de ferramentas de monitoramento, foi constatado que quando se alcançou uma resolução intermediária, no caso a de 1152 x 648, o ganho de acurácia obtido foi mínimo e não justificava o custo computacional necessário que teria de ser investido para processar as imagens.

Através destes resultados, verifica-se também que, dentre as variáveis observadas, a memória RAM do computador é a que se mostrou mais requisitada para a execução do algoritmo. Também verificou-se a eficácia do algoritmo adotado: SubSENSE, que obteve uma acurácia de 79 % no *dataset* Cornelio4k.

Assim, espera-se que o desenvolvimento deste estudo, juntamente com seus resultados sirvam de referência na área de processamento de imagens no que se refere à segmentação de imagens de alta resolução. Alguns trabalhos futuros que podem ser abordados são: expansão do *dataset*, tendo em vista que com um número maior de imagens seria possível obter-se resultados mais precisos; desenvolver um novo estudo utilizando outros algoritmos de segmentação de modo a observar em qual resolução há maior ganho.