

UNIVERSIDADE DO ESTADO DO AMAZONAS – UEA  
ESCOLA SUPERIOR DE TECNOLOGIA - EST  
CURSO DE ENGENHARIA ELÉTRICA

LEANDRO DE SOUZA BEZERRA

**UM ESTUDO COMPARATIVO DA EFICIÊNCIA DE ALGORITMOS DE  
SUBTRAÇÃO DE FUNDO DE IMAGEM APLICADOS A UM AMBIENTE  
SIMULADO DE TRÁFEGO VEICULAR**

Manaus

2023

LEANDRO DE SOUZA BEZERRA

**UM ESTUDO COMPARATIVO DA EFICIÊNCIA DE ALGORITMOS DE  
SUBTRAÇÃO DE FUNDO DE IMAGEM APLICADOS A UM AMBIENTE  
SIMULADO DE TRÁFEGO VEICULAR**

Projeto de Pesquisa desenvolvido durante a disciplina de Trabalho de Conclusão de Curso II e apresentada à banca avaliadora do Curso de Engenharia Elétrica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como pré-requisito para obtenção do título de Engenheiro Eletricista.

Orientador: Prof. Dr. Fábio de Sousa Cardoso.

Manaus

2023

**Universidade do Estado do Amazonas – UEA**  
**Escola Superior de Tecnologia - EST**

*Reitor:*

**André Luiz Nunes Zogahib**

*Vice-Reitor:*

**Kátia do Nascimento Couceiro**

*Diretora da Escola Superior de Tecnologia:*

**Jucimar Maia da Silva Júnior**

*Coordenador do Curso de Engenharia Elétrica:*

**Israel Gondres Torné**

*Banca Avaliadora composta por:*

**Prof. Fábio de Sousa Cardoso, Dr. (Orientador)**

**Prof. Angilberto Muniz Ferreira Sobrinho, Dr.**

**Prof. Antônio Luiz Alencar Pantoja, Dr.**

*Data da defesa: 12/09/2023.*

## **CIP – Catalogação na Publicação**

Bezerra, Leandro de Souza

Um estudo comparativo da eficiência de algoritmos de subtração de fundo de imagem aplicados a um ambiente simulado de tráfego veicular / Leandro de Souza Bezerra; [orientado por] Fábio de Sousa Cardoso. – Manaus: 2023.

101 p.: il.

Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica). Universidade do Estado do Amazonas, 2023.

1. Subtração de Fundo. 2. Processamento Digital de Imagens. 3. Visão Computacional. 4. Detecção de Objetos. I. Cardoso, Fábio de Sousa.

LEANDRO DE SOUZA BEZERRA

**UM ESTUDO COMPARATIVO DA EFICIÊNCIA DE ALGORITMOS DE  
SUBTRAÇÃO DE FUNDO DE IMAGEM APLICADOS A UM AMBIENTE  
SIMULADO DE TRÁFEGO VEICULAR**

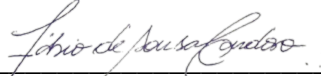
Pesquisa desenvolvida durante a disciplina de Trabalho de Conclusão de Curso II e apresentada à banca avaliadora do Curso de Engenharia Elétrica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como pré-requisito para a obtenção do título de Engenheiro Eletricista.

Nota obtida: 9,9 (Nove Vírgula Nove)

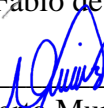
Aprovada em 12 / 09 / 2023.

Área de concentração: Processamento Digital de Imagens

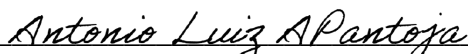
BANCA EXAMINADORA



\_\_\_\_\_  
Orientador: Fábio de Sousa Cardoso, Dr.



\_\_\_\_\_  
Avaliador: Angilberto Muniz Ferreira Sobrinho, Dr.



\_\_\_\_\_  
Avaliador: Antônio Luiz Alencar Pantoja, Dr.

Manaus

2023

*Dedico esta conquista aos meus pais, como forma de gratidão, por todo o suporte, auxílio e pela prioridade que proporcionaram ao meu desenvolvimento e ao da minha irmã, especialmente em nossa educação. Eles foram fundamentais para este momento tão importante da minha vida.*

## **AGRADECIMENTOS**

Agradeço primeiramente à Deus, pela vida, pela saúde e por guiar-me em minha trajetória.

Agradeço a todos os meus familiares, especialmente aos meus pais, que sempre foram exemplares e estiveram ao meu lado apoiando-me em todos os momentos da minha vida.

Ao meu orientador e todos os professores que me auxiliaram durante este trabalho e que contribuíram para o meu desenvolvimento acadêmico durante toda graduação.

A todos os meus amigos da faculdade e do trabalho, os quais me deram apoio e que contribuíram em minha vida acadêmica.

## RESUMO

Este trabalho tem como objetivo realizar o estudo do método de detecção de objetos denominado Subtração de Fundo (*Background Subtraction*), analisar o desempenho das técnicas de modelagem MOG, MOG2, GMG, KNN, CNT quando aplicado em um ambiente simulado de tráfego veicular e identificar a abordagem com desempenho mais adequado para este tipo de situação. A metodologia utilizada consiste no desenvolvimento de um algoritmo que irá implementar cada técnica de modelagem em um vídeo que apresenta veículos em movimento, onde os resultados da subtração serão comparados com cada um dos *pixels* das imagens de referência, as quais representam o resultado ideal (*Ground Truth*). Os dados coletados a partir das comparações são utilizados para o cálculo das métricas de acurácia e tempo de processamento, sendo estes os indicadores utilizados para análise de desempenho e seleção da técnica de modelagem. A partir dos resultados do experimento constatou-se que a técnica MOG2 é a mais apropriada para o cenário proposto, apresentando a melhor precisão (66,16%), exatidão (98,59%) e com um tempo médio de processamento por *frame* adequado (8,15 ms) quando comparado com as demais técnicas analisadas.

**Palavras-chave:** Subtração de Fundo, Processamento Digital de Imagens, Visão Computacional, Detecção de Objetos.

## ABSTRACT

The main objective of this work is to conduct a research related to Background Subtraction object detection method and analyze its performance when applying MOG, MOG2, GMG, KNN and CNT modeling techniques in a simulated vehicular traffic environment and identify the approach with the most suitable performance for this type of situation. The methodology used consists of developing an algorithm that will implement each modeling technique in a video that presents moving vehicles, where the results obtained by the subtraction will be compared with each pixels from the reference images, which represents the ideal result (Ground Truth). The data collected from the comparisons are used to calculate the accuracy and processing time metrics, which are the indicators used for performance analysis and selection of the modeling technique. The results of the experiment demonstrated that MOG2 technique is the most appropriate for the proposed scenario, with the best precision (66.16%), accuracy (98.59%) and suitable average processing time per frame (8.15 ms) when compared to the other techniques evaluated.

**Keywords:** Background Subtraction, Digital Image Processing, Computer Vision, Object Detection.



# LISTA DE ILUSTRAÇÕES

## FIGURAS

Figura 1 – Ilustração de aplicações de sistemas de transporte inteligente.....	18
Figura 2 - Exemplo de diagrama de estágios.....	20
Figura 3 - Exemplo de diagrama de intervalos luminosos .....	20
Figura 4 - Representação de zonas de monitoramento de presença veicular .....	23
Figura 5 - Componentes de um sistema de processamento de imagens de uso geral.....	24
Figura 6 - Matriz para representação matemática de uma imagem digital.....	26
Figura 7 - Representação de uma imagem digital bidimensional.....	26
Figura 8 - Sensor de aquisição de imagens (unidade) .....	28
Figura 9 - Sensores de área (matricial).....	28
Figura 10 - Processo de aquisição de uma imagem digital .....	29
Figura 11 - Etapas para produção de uma imagem digital .....	30
Figura 12 - Imagem contínua projetada em uma matriz de sensores (esquerda) em comparação ao resultado da digitalização da imagem após amostragem e quantização (direita) .....	31
Figura 13 - Exemplo de imagem policromática (colorida) e em escala de cinzas .....	32
Figura 14 - Cores primárias e secundárias de luz e pigmentos .....	32
Figura 15 - Modelo RGB.....	33
Figura 16 - Vizinhança/Máscara 3x3 em um ponto de coordenadas (x,y) de uma imagem ...	34
Figura 17 - Máscaras para cálculo do filtro de média: a) 3x3 , b) 5x5, c) 7x7.....	35
Figura 18 - Utilização do filtro de média para efeito de desfoque .....	35
Figura 19 - Utilização do filtro de mediana para remoção de ruído (sal-e-pimenta) .....	36
Figura 20 - Utilização do filtro de realce para melhoria de nitidez.....	37
Figura 21 - Máscaras para implementação de realce por Laplaciano.....	38
Figura 22 - Efeito da utilização de um limiar para binarização de uma imagem .....	38
Figura 23 - Representação do processo de dilatação de uma imagem .....	41
Figura 24 - Representação do efeito da operação de dilatação.....	42
Figura 25 - Representação do processo de dilatação de uma imagem .....	43
Figura 26 - Representação do efeito da operação de erosão.....	43
Figura 27 - Representação das operações morfológicas de abertura e fechadura .....	44
Figura 28 - Reconhecimento de objetos a partir de visão computacional .....	46
Figura 29 - Exemplo de aplicação de visão computacional para detecção veicular.....	46

Figura 30 - Diferença entre imagens realizada por subtração de fundo .....	48
Figura 31 - Diagrama geral de um algoritmo de subtração de fundo .....	51
Figura 32 - Diagrama de blocos do algoritmo GMG.....	52
Figura 33 - Classificação de dados por KNN .....	54
Figura 34 - Subtração de fundo por mistura de gaussianas .....	57
Figura 35 - Distribuições Gaussianas de probabilidade do valor de um pixel .....	60
Figura 36 - Diagrama representativo do algoritmo utilizado para implementação da subtração de fundo .....	66
Figura 37 - Representação de uma imagem de referência (Ground Truth) .....	67
Figura 38 - Tabela de contingência utilizada para o teste de acurácia .....	68
Figura 39 - Comparação entre Ground Truth e imagem segmentada por KNN.....	69
Figura 40 - Diagrama do algoritmo utilizado para teste de acurácia (k = 23, ID[k] = 3933)...	71
Figura 41 - Diagrama do algoritmo utilizado para teste de tempo de processamento.....	72
Figura 42 - Amostra de <i>frame</i> do vídeo utilizado para o projeto .....	74
Figura 43 - Exemplo de <i>frame</i> capturado durante o processamento do vídeo .....	74
Figura 44 - Conversão do <i>frame</i> capturado de RGB para escala de cinzas.....	75
Figura 45 - <i>Frame</i> resultante após filtragem espacial .....	75
Figura 46 - Imagem binarizada após aplicação de subtração de fundo (MOG) .....	76
Figura 47 - Resultado após aplicação de operações morfológicas .....	77
Figura 48 - <i>Frames</i> do vídeo de entrada e imagem após processamento por subtração de fundo .....	77

## GRÁFICOS

Gráfico 1 - Taxa de Verdadeiros Positivos (TVP) (%) .....	82
Gráfico 2 - Taxa de falsos positivos (TFP) (%).....	82
Gráfico 3 - Negativos corretamente detectados - N (%) .....	83
Gráfico 4 - Taxa de precisão (%).....	84
Gráfico 5 - Taxa de exatidão (%) .....	84
Gráfico 6 - Tempo médio requisitado para o processamento de 1500 <i>frames</i> .....	86
Gráfico 7 - Tempo médio de processamento por <i>frame</i> .....	87

## LISTA DE TABELAS

Tabela 1 - Resultados da segmentação das imagens por subtração de fundo.....	80
Tabela 2 - Resultados do teste de acurácia .....	81
Tabela 3 - Resultados do tempo médio total de processamento para 1500 <i>frames</i> .....	85
Tabela 4 - Resultados do tempo médio de processamento por <i>frame</i> .....	86

## LISTA DE ABREVIATURAS E SIGLAS

BSM - *Background Subtraction Method*

MoG – *Mixture of Gaussians*

GMG – Godbehere, Matsukawa, Goldberg

CNT – CouNT

KNN – *K-Nearest Neighbors*

CONTRAN – Conselho Nacional de Trânsito

SENATRAN – Secretaria Nacional de Trânsito

ITS – *Intelligent Transportation Systems*

SCATS – *Sydney Coordinated Adaptive Traffic System*

SURTRAC – *Scalable Urban Traffic Control*

ATMS – *Advanced Traffic Management System*

CCD – *Charged Coupled Device*

SVC – Sinal de Vídeo Composto

RGB – *Red, Green and Blue*

ES – Elemento Estruturante

OpenCV – *Open Source Computer Vision Library*

HOG – *Histogram of Oriented Gradients*

R-CNN – *Region-Based Convolutional Neural Network*

YOLO – *You Only Look Once*

IDE – *Integrated Development Environment*

RAM – *Random Access Memory*

VP – Verdadeiro Positivo

FP – Falso Positivo

FN – Falso Negativo

VN – Verdadeiro Negativo

TVP – Taxa de Verdadeiro Positivo

TFP – Taxa de Falso Positivo

s. – Segundos

ms. - Milissegundos

## SUMÁRIO

<b>INTRODUÇÃO</b> .....	14
<b>1 REFERENCIAL TEÓRICO</b> .....	16
1.1 CIDADES INTELIGENTES .....	16
1.1.1 Sistemas Inteligentes de Transporte .....	17
1.2 CONTROLE E SINALIZAÇÃO SEMAFÓRICA .....	19
1.3 FUNDAMENTOS DE PROCESSAMENTO DIGITAL DE IMAGENS.....	23
1.3.1 Fundamentos de Imagem.....	25
1.3.2 Aquisição de Imagem .....	27
1.3.3 Amostragem e Quantização .....	29
1.3.4 Imagens Monocromáticas e Policromáticas .....	31
1.3.5 Filtragem Espacial .....	33
1.3.6 Limiarização .....	38
1.3.7 Morfologia .....	39
1.3.7.1 Dilatação .....	41
1.3.7.2 Erosão .....	43
1.3.7.3 Abertura e fechamento .....	45
1.4 VISÃO COMPUTACIONAL .....	45
1.5 SUBTRAÇÃO DE FUNDO DE IMAGEM .....	47
1.5.1 Método de Godbehere, Matsukawa, Goldberg (GMG).....	51
1.5.2 <i>K-nearest Neighbors</i> (KNN).....	53
1.5.3 <i>Count</i> (CNT).....	55
1.5.4 Mistura de Gaussianas (MOG/MOG2).....	56
1.6 <i>OPENCV</i> .....	61
<b>2 METODOLOGIA</b> .....	62
2.1 PREPARAÇÃO DO AMBIENTE DE DESENVOLVIMENTO.....	63
2.2 DESENVOLVIMENTO DO ALGORITMO DE SUBTRAÇÃO DE FUNDO .....	64
2.3 COLETA DE DADOS PARA MÉTRICAS DE DESEMPENHO .....	66
<b>3 IMPLEMENTAÇÃO</b> .....	73
3.1 SELEÇÃO DO VÍDEO .....	73
3.2 IMPLEMENTAÇÃO DO ALGORITMO DE SUBTRAÇÃO DE FUNDO .....	74
3.3 TESTES DE ACURÁCIA.....	78
3.4 TESTES DE TEMPO DE PROCESSAMENTO .....	78

<b>4</b>	<b>RESULTADOS E DISCUSSÕES</b> .....	80
4.1	RESULTADOS DO TESTE DE ACURÁCIA .....	81
4.2	RESULTADOS DO TEMPO DE PROCESSAMENTO .....	85
	<b>CONCLUSÃO</b> .....	89
	<b>REFERÊNCIAS</b> .....	91
	<b>APÊNDICE A – CÓDIGO PARA IMPLEMENTAÇÃO DE SUBTRAÇÃO DE FUNDO</b> 94	
	<b>APÊNDICE B – CÓDIGO PARA TESTES DE <i>GROUND TRUTH</i></b> .....	97
	<b>APÊNDICE C – CÓDIGO PARA TESTE DE TEMPO DE PROCESSAMENTO</b> .....	99

## INTRODUÇÃO

O crescimento populacional em grandes cidades é uma característica evidente da sociedade moderna, fator que traz a necessidade de adotar estratégias inovadoras para solucionar desafios que surgem a partir desse cenário, tal como o aumento dos congestionamentos ocasionados pelo elevado crescimento da frota de veículos em regiões urbanas. Segundo a Secretaria Nacional de Trânsito - SENATRAN, nas estatísticas calculadas entre os anos de 2011 e 2021 na cidade de Manaus, houve um aumento de 64,56% na frota de veículos na cidade. Este crescimento possibilita a ocorrência de impactos negativos na cidade, como por exemplo, na ocorrência de congestionamentos, aumento no tempo de viagem, poluição e redução nos índices de bem-estar social.

Para melhoria desta situação, é necessário propor avanços a partir da implementação de novas alternativas tecnológicas para o aperfeiçoamento do gerenciamento de tráfego, em contraste ao crescimento contínuo da quantidade de automóveis em grandes cidades. Tais resultados podem ser alcançados com sistemas semafóricos inteligentes, por exemplo, que utilizam tecnologias de visão computacional.

O presente trabalho visa realizar um estudo acerca do método de subtração de fundo de imagens, abordagem comumente utilizada na área de visão computacional para a detecção de objetos. O objetivo é apresentar a técnica de subtração de fundo mais adequada em um cenário de aplicação voltada ao tráfego veicular, analisando e comparando as métricas de desempenho (acurácia e processamento) dos algoritmos MOG, MOG2, GMG, KNN e CNT, as quais executam este método, quando aplicados em um ambiente simulado com movimentação de veículos.

A pesquisa foi ordenada em quatro capítulos, as quais são: referencial teórico, metodologia, implementação e resultados e discussões.

Capítulo 1 – Referencial teórico: Neste capítulo são apresentados inicialmente os conceitos de cidades inteligentes e sistemas inteligentes de trânsito, demonstrando a aplicabilidade e a tendência da implementação de novas tecnologias na infraestrutura de trânsito das cidades, estabelecendo um enfoque nos sistemas semafóricos. Em seguida, são apresentados os conceitos fundamentais do processamento digital de imagens, o conceito geral e suas principais operações. Por fim, é feita a descrição do conceito de subtração de fundo de imagens e das técnicas de modelagem de plano de fundo selecionadas como objeto de estudo.

Capítulo 2 – Metodologia: É o capítulo em que são descritos os processos de desenvolvimento do programa que irá implementar cada uma das técnicas de modelagem de

plano de fundo e a apresentação das métricas que serão utilizadas para realização da análise dos resultados.

Capítulo 3 – Implementação: Neste item é feita a descrição do procedimento prático realizado havendo como base a metodologia estabelecida.

Capítulo 4 – Resultados e discussões: Neste capítulo é feita a exposição e discussão dos resultados obtidos a partir dos testes realizados, baseando-se no conhecimento teórico relacionado às técnicas de modelagem de plano de fundo abordados. A análise das métricas obtidas são feitas considerando um cenário de aplicação em um sistema de detecção de veículos em movimento.



## 1 REFERENCIAL TEÓRICO

Apresentado o problema de pesquisa e a respectiva proposta, faz-se necessário descrever a respeito dos assuntos que envolvem o tema abordado. A princípio, é discorrido a respeito da temática de trânsito, abordando os conceitos de tecnologia que a envolvem e o que diz a norma brasileira em relação a aplicações que utilizam sistemas de visão, com o intuito de apresentar o cenário atual referente à novas tecnologias de controle de tráfego. Em seguida, são apresentados os principais fundamentos relacionados ao processamento digital de imagens, ao qual forma a base para aplicações de visão computacional. A fundamentação teórica é finalizada ao realizar descrição do método de subtração de imagens, suas características e as principais técnicas de modelagem selecionadas como objeto de estudo.

### 1.1 CIDADES INTELIGENTES

O conceito de Cidades Inteligentes, ou *Smart Cities*, apresenta a aplicação de tecnologias para coleta e gerenciamento de dados para a implementação de uma cidade inteligente, conectada, eficiente e sustentável. O termo foi adotado pela primeira vez por grandes empresas como Siemens, Cisco e IBM, desde o início do século XXI, para se referir à aplicação de sistemas de informação complexos para integrar a operação de infraestruturas urbanas e de serviços, como edifícios, redes de transporte, eletricidade, distribuição de água e segurança pública, dentre outros fatores principais em uma sociedade. Desde então, evolui para significar quase qualquer forma de inovação baseada em tecnologia no planejamento, desenvolvimento e, por exemplo, na implantação de serviços para veículos elétricos plug-in.

A aplicação de tecnologia de informação em cidades inteligentes tem como proposta trazer benefícios em diversos aspectos da sociedade, como por exemplo (HARRISON et al, 2011, tradução nossa):

- a) Reduzir o consumo de recursos, tais como energia e água.
- b) Contribuir com a redução de emissão de gás carbônico.
- c) Desenvolver novos serviços para acesso da população, como por exemplo orientação em tempo real sobre a melhor forma de explorar as modalidades de transporte disponíveis.
- d) Prover melhorias para empresas, através de publicações de informação em tempo real na operação de serviços da cidade.

- e) Revelando como as demandas por energia, água e transporte atingem o pico na escala da cidade para que os gestores da cidade possam colaborar para suavizar esses picos e melhorar a resiliência

“As principais características de uma *Smart City* são: *Smart economy*, relacionada à capacidade de inovação, produtividade, competitividade econômica e a sua flexibilidade de mercado; *Smart People*, que tem suas bases relacionadas à qualificação dos seus cidadãos, na qualidade de bem-estar social, interações sociais e na sua capacidade de abertura a relações exteriores; *Smart Governance*, que tem como principais características a qualidade do funcionamento da administração, a participação política e os serviços oferecidos aos cidadãos; *Smart Mobility*, que tem como pilares a acessibilidade da cidade, bem como a utilização de sistemas de transporte modernos e sustentáveis; *Smart Environment*, que trata da administração dos seus recursos e da proteção do meio ambiente; e *Smart Living*, que relaciona diversos aspectos de qualidade de vida, saúde, segurança e turismo de uma cidade.” (FAISCA et al, 2019, p.2).

### 1.1.1 Sistemas Inteligentes de Transporte

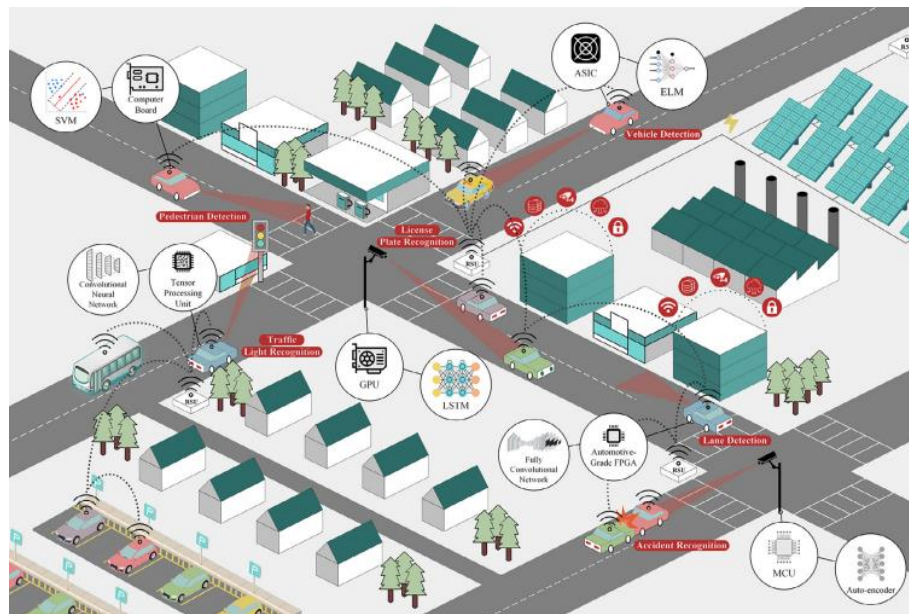
O conceito de Sistemas Inteligentes de Transporte (*Intelligent Transportation Systems - ITS*), é definida como a aplicação de computação, informação e tecnologias de comunicação para o gerenciamento em tempo real de veículos e redes que envolvem locomoção de pessoas, materiais e serviços (SAMADI et al, 2012). Dentro deste conceito, estão os Sistemas de Gerenciamento Avançado de Tráfego (*Advanced Traffic Management System – ATMS*), que são sub sistemas de maior proporção, com o objetivo de implementar sistemas de tráfego adaptativo a partir de tecnologias avançadas em comparação aos sistemas de controle de tráfego tradicionais. Os objetivos da utilização deste tipo de sistema estão na redução de congestionamentos, na melhoria de fluxo de veículos e pessoas, gerenciamento de demanda de tráfego e definição de prioridades.

Os sistemas ITS têm sido caracterizados como um tema de grande interesse em pesquisa por ser considerado um tema promissor para a reestruturação no setor de transporte, a partir de novas tecnologias que podem ser desenvolvidas para a solução de problemas comuns relacionados à de tráfego de veículos. Esse tipo de sistema compreende uma multiplicidade de feitos de engenharia que funcionam como uma entidade para otimizar experiências de viagem

em uma escala de rede a partir de um aspecto técnico, social, econômico e ambiental (DAMAJ et al, 2021).

A figura 1 ilustra o conceito de aplicações inteligentes em um cenário de trânsito, onde a partir do avanço nos conhecimentos de inteligência artificial e visão computacional, há o surgimento de tecnologias como os de sistemas de sinalização inteligentes, veículos autônomos, dentro outros.

Figura 1 – Ilustração de aplicações de sistemas de transporte inteligente



Fonte: Damaj et al (2021)

Sistemas de controle de tráfego adaptativo fazem um melhor uso da capacidade geral de tráfego de uma via e melhoram a eficiência da rede de tráfego, coletando dados que podem ser utilizados para melhoria contínua de fluxo da região. A aplicação deste tipo de tecnologia pode ser encontrada em países como a Austrália, que implementa o sistema de SCATS (*Sydney Coordinated Adaptive Traffic System*) que opera a partir de uma seleção automática de planos semafóricos pré-estabelecidos em uma biblioteca de armazenamento local e que são selecionados de acordo com os dados obtidos por meio de sensores (Sensores indutivos, e. g.) que são estabelecidos em cada semáforo da via, realizando a detecção de veículos e pedestres. O sistema SURTRAC (*Scalable Urban Traffic Control*) também é um outro exemplo de sistema de controle de tráfego adaptativo, com origem nos Estados Unidos. Este sistema realiza o monitoramento de tráfego em tempo real, é composto por câmeras que são utilizadas como sensores para aquisição de dados de tráfego e detecção de objetos, cujo essas informações são

processadas por meio de uma API e são enviadas para o controlador do sistema que envia os comandos para os semáforos participantes da rede de gerenciamento de trânsito na região, definindo os planos de fases do próprio semáforo e dos sinais vizinhos (SAMADI et al, 2012; SMITH et al, 2013).

## 1.2 CONTROLE E SINALIZAÇÃO SEMAFÓRICA

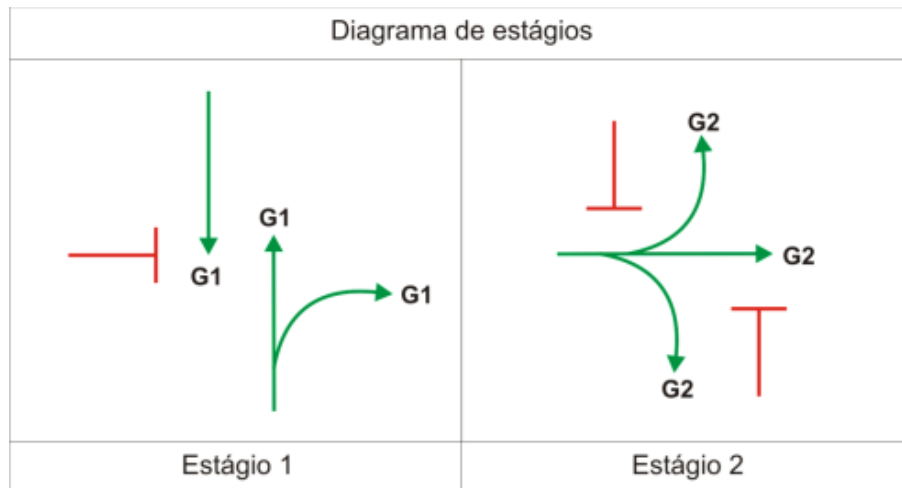
O objetivo principal dos semáforos está em organizar o tráfego de veículos e pessoas, evitando a ocorrência de acidentes que podem causar danos materiais em veículos ou físicos em pedestres. Os mesmos itens podem ser utilizados para a melhoria de fluxo de veículos em uma via, estabelecendo uma programação semafórica que atenda ao perfil de trânsito do local. O uso incorreto destes dispositivos pode provocar o aumento de congestionamentos, atrasos e acidentes. Assim, o estabelecimento de uma programação semafórica ideal é importante para a melhoria de eficiência de tráfego local.

Em um sistema semafórico, caracterizam-se os seguintes conceitos básicos:

- a) **Grupo de movimentos:** Conjuntos de movimentos possíveis presentes em uma mesma aproximação e que permite o direito de passagem simultaneamente.
- b) **Grupo semafórico:** Conjuntos de semáforos (grupos focais) com indicações luminosas idênticas utilizadas para o controle de grupos de movimentos que recebem simultaneamente o direito de passagem.
- c) **Plano semafórico:** Conjunto de elementos que caracterizam a programação da sinalização semafórica para uma interseção ou seção de via, em um determinado período do dia.
- d) **Ciclo:** É o tempo necessário para que ocorra uma sequência completa de indicações luminosas (sinalizações vermelha, amarela e verde) de todos os estágios de uma sinalização semafórica. É definida pela soma dos tempos de todos os estágios programados para o controle do tráfego local.
- e) **Estágio:** Intervalo de tempo em que um ou mais grupos de movimentos recebem simultaneamente o direito de passagem. O estágio compreende o tempo de verde e o tempo de entreverdes que o segue
- f) **Entreverdes:** intervalo de tempo entre o final do verde de um estágio e o início do verde do estágio subsequente, no qual ocorre a alteração do conjunto de movimentos autorizados e bloqueados.

- g) **Defasagem (Offset):** É a diferença em segundos entre os verdes de dois sinais de uma mesma via, utilizado para sincronização de semáforos em um fluxo de mesmo sentido.
- h) **Fase:** conjunto de movimentos comandados por uma mesma sequência de indicações luminosas nos estágios do ciclo.
- i) **Diagrama de estágios:** Representação gráfica da alocação dos movimentos que podem ser realizados em cada estágio do ciclo, conforme ilustrado na figura 2.
- j) **Diagrama de intervalos luminosos:** Representação gráfica da duração e sequência dos intervalos luminosos e estágios associados aos grupos semafóricos correspondentes, conforme ilustrado na figura 3.

Figura 2 - Exemplo de diagrama de estágios



Fonte: CONTRAN (2014)

Figura 3 - Exemplo de diagrama de intervalos luminosos

DIAGRAMA DE INTERVALOS LUMINOSOS						
INSTANTE (s)	0	30	34	36	74	78
G1	[Barra Verde]		[Barra Amarela]	[Barra Vermelha]	[Barra Vermelha]	[Barra Vermelha]
G2	[Barra Vermelha]		[Barra Vermelha]	[Barra Verde]	[Barra Verde]	[Barra Verde]
INTERVALOS	1	2	3	4	5	6
DURAÇÃO (s)	30	4	2	38	4	2
% CICLO	38	5	2	48	5	2
ESTÁGIOS	1			2		

Fonte: CONTRAN (2014)

A programação de fases é estabelecida por parâmetros de tempo específicos que devem ser calculados de acordo com o padrão de fluxo local, possibilitando a escolha de uma programação favorável à situação. A teoria inicial de cálculo para programação semafórica, considerando um semáforo isolado, utiliza-se de equações básicas de tempo de ciclo, tempos de verde (sinalização verde) e defasagem. Parâmetros de tempo de sinal em cada movimento de uma via específica devem ser analisados para calcular os tempos de ciclo. O tempo de transição de estágios do sinal vermelho entre os dois movimentos também é um fundamental para o cálculo do tempo de ciclo.

Segundo o Conselho Nacional de Trânsito – CONTRAN (2014) em Manual Brasileiro de Sinalização de Trânsito – Volume V: Sinalização Semafórica, existem dois tipos de controle que podem ser aplicados em sinalizações semafóricas: Controle em tempo fixo e controle atuado pelo tráfego. A diferença entre os dois tipos está na resposta do sistema a variações de tráfego no local.

O controle em tempo fixo utiliza planos semafóricos calculados com base nas estatísticas coletadas relacionadas ao padrão de fluxo de veículos do local em que o sistema está sendo aplicado, obtidos por meio contagem volumétrica e outros tipos de levantamentos de campo. O controle pode ser estabelecido para um único plano semafórico, ou na adoção de planos específicos para diferentes períodos do dia e da semana, aplicados de acordo com a demanda de tráfego em momentos distintos. Em finais de semana, por exemplo, o fluxo de veículos pode tender a ser baixo, ao contrário do padrão que pode ser presenciado em um dia da semana pelo horário da tarde, quando há grande movimentação de veículos. Nesse tipo de controle, parâmetros como tempo de ciclo, sequência de estágios, duração dos intervalos luminosos e a defasagem são fixos, sendo alterados somente a partir da ação de um profissional de trânsito.

O controle atuado pode ser caracterizado de duas formas: semiatuado ou totalmente atuado.

O controle semiatuado é em geral aplicado em cruzamentos de vias de grande volume (principais) com vias de baixo volume de tráfego (secundárias). Esse tipo de controle estabelece como prioridade o fluxo de veículos na via principal, mantendo continuamente a sinalização verde para o movimento nesta via. Caso os detectores implementados no sistema identificarem a presença de veículos motorizados na via secundária, a sinalização verde passa a ser interrompida para o movimento na via principal. Para o caso de travessia de pedestres e bicicletas na via, é empregado o uso de botoeiras que quando acionadas interrompem a indicação verde para os veículos da via principal.

O controle totalmente atuado decorre do monitoramento da demanda de tráfego na interseção, mediante a implantação de detectores de tráfego em todas as suas aproximações, permitindo alterações nos tempos dos estágios. O princípio básico de um controlador totalmente atuado consiste em determinar valores pré-estabelecidos de tempos máximos e mínimos de sinalização verde associados a cada estágio, e que deve variar de acordo com a demanda de veículos local para cada via. Esse tipo de controle permite o ajuste automático e em tempo real dos valores de determinados parâmetros de programação, como a prioridade a ser dada para uma região congestionada, por exemplo.

Sistemas de controle atuado precisam de sensores para compor o sistema que irá implementar o controle semaforico. Esses equipamentos realizam aquisição de informações do ambiente respondendo à estímulos físicos, e devem ser utilizados nessa situação para a identificação de presença de veículos. É a partir do uso deste componente que o sistema poderá detectar e atuar de acordo com o tráfego. Dentre as principais abordagens utilizadas para detecção de tráfego para implementação de controle atuado, estão: Laços detectores indutivos; Laços virtuais por tratamento de imagem; Detectores por micro-ondas; Detecção magnética; Detecção por radiação infravermelha e detecção ultrassônica.

A utilização de laços virtuais ou detecção de tráfego veicular por tratamento de imagem é uma abordagem que pode trazer vantagens em relação a outros tipos de métodos. A utilização de câmeras para aquisição de imagens e detecção permite coletar informações adicionais e precisas em comparação a outros tipos de abordagens, por exemplo, identificando os tipos de veículos que trafegam na via. Em comparação aos sensores implementados a partir de laços indutivos, cujo é a abordagem mais utilizada, os laços virtuais por tratamento de imagem têm como vantagem a manutenção, instalação e eficiência contra falhas. As bobinas utilizadas para composição de laços indutivos precisam ser inseridas na pavimentação da via em que será implementado o sistema de controle atuado, o que acarreta a possibilidade de falhas e dificulta sua manutenção pois é necessário realizar a remoção da pavimentação para a inserção dos laços indutivos. Assim, a utilização de câmeras pode ser considerada uma boa opção tendo em vista as vantagens proporcionadas comparando-se a praticidade para implementação e sua eficiência.

A estratégia de controle deve ser considerada para um sistema de detecção semaforica, onde deve ser determinado se a estratégia de implementação será isolada, caso em que cada interseção é controlada independentemente das demais vias próximas ao local, ou controlada em rede, estratégia que visa a melhoria de circulação de tráfego em uma via ou percurso pré-estabelecido em uma rede fechada (conhecido como controle em corredor). Para o caso de

controle em rede aberta, a estratégia busca a melhoria de desempenho de tráfego em uma determinada região, denominando-se controle de área.

A figura 4 exemplifica a utilização de um detector de presença veículos a partir de visão computacional, denominados como laços virtuais por tratamento de imagens.

Figura 4 - Representação de zonas de monitoramento de presença veicular



Fonte: CONTRAN (2014)

### 1.3 FUNDAMENTOS DE PROCESSAMENTO DIGITAL DE IMAGENS

O processamento digital de imagens refere-se ao campo de estudo em que a partir de um computador digital, é realizado a aplicação de operações e demais processamentos para realizar o tratamento de uma imagem digitalizada. Esta área de estudo é caracterizada por tratar imagens tanto na entrada quanto na saída de um processo. As operações realizadas em um processamento digital de uma imagem podem resultar em melhorias dentre as quais podem ser citadas: ajustes de brilho, equalização, melhoria de nitidez, contraste de uma imagem, remoção de ruídos, dentre outros exemplos.

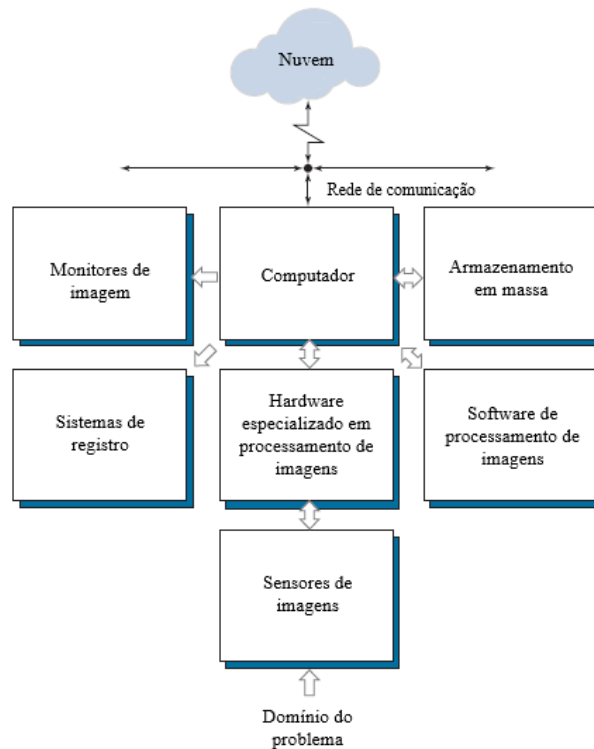
O estudo relacionado ao processamento digital de imagens possibilitou o surgimento de melhorias em diversas áreas da sociedade, dentre as quais é possível citar a área médica, onde encontram-se equipamentos de ultrassom e raio X, que quando adquiridos em suas versões mais atuais, utilizam tecnologias relacionadas à esta área, permitindo ao profissional de medicina realizar sua análise e diagnóstico de maneira simplificada e em tempo reduzido quando comparado aos métodos tradicionais proporcionados por equipamentos de gerações anteriores.



Os elementos de um sistema de processamento de imagens de uso genérico devem abranger as principais operações que podem ser efetuadas sobre uma imagem, a saber: aquisição, armazenamento, processamento e exibição. (MARQUES; VIEIRA, 1999).

Segundo Gonzalez e Woods (2018), um sistema de processamento de imagem tem como seus principais componentes: sensores de imagem, um *hardware* especializado em processamento de imagens, computador, *software* de processamento de imagens, sistema de registro, um sistema de armazenamento e monitores de imagem, conforme ilustrado na figura 5.

Figura 5 - Componentes de um sistema de processamento de imagens de uso geral



Fonte: Gonzalez e Woods (2018)

Existem diversos campos de estudos similares à área de processamento digital de imagens, tais como a área de análise de imagens e visão computacional. Apesar de não haver uma forma clara e objetiva de diferenciar tais campos, considera-se que o processamento digital de uma imagem é uma etapa fundamental para ambas as áreas. Por exemplo, em uma aplicação de visão computacional cujo o algoritmo deve reconhecer frutas em um banco de imagens, identificar seus respectivos nomes e seu estado de conservação, é evidente que o mesmo deve conter em seu sistema uma etapa de pré-processamento para permitir a realização desta

operação. Além disso, processos adicionais podem ser incluídos no tratamento das imagens permitindo a melhoria da eficiência do algoritmo em reconhecer objetos.

Um paradigma útil para a diferenciação entre os campos de processamento digital de imagens, análise de imagens e visão computacional seria organizá-los em uma linha contínua dividindo-os em diferentes níveis de processamento: baixo, médio e alto. Um processamento de imagens de nível baixo é caracterizado pelo fato de tanto a entrada quanto a saída serem imagens. Enquanto isso, o processamento de imagens de nível médio é caracterizado pelo fato de suas entradas, em geral, serem imagens, mas as saídas são atributos extraídos desta imagem (bordas, contornos e identidade de objetos individuais, por exemplo), envolvendo tarefas como a segmentação (separação de uma imagem em regiões ou objetos), a descrição desses objetos para reduzi-los a uma forma adequada para o processamento computacional e a classificação (reconhecimento) de objetos individuais. Por fim, o processamento de nível alto envolve “dar sentido” a um conjunto de objetos reconhecidos (como ocorre no campo de análise de imagens) e realizar funções cognitivas normalmente associadas à visão (visão computacional). (GONZALEZ; WOODS, 2018).

### 1.3.1 Fundamentos de Imagem

A formação de uma imagem ocorre quando um sensor registra a radiação eletromagnética (luz, por exemplo) interagida em objetos físicos (BALLARD; BROWN, 1982, tradução nossa). Existem três componentes básicos para a formação de uma imagem: a iluminação, os modelos de reflectância das superfícies que são visualizadas e o processo de formação da imagem na retina dos olhos humanos ou no sensor de uma câmera (ACHARYA; RAY, 2005, tradução nossa).

Uma imagem pode ser definida e representada matematicamente como uma função bidimensional,  $f(x,y)$ , assumindo um formato de matriz conforme ilustra a figura 6, onde  $x$  e  $y$  são coordenadas espaciais (plano), e a amplitude de  $f$  em qualquer par de coordenadas  $(x,y)$  é chamada de intensidade ou nível de cinza da imagem nesse ponto. Quando  $x,y$  e os valores da intensidade de  $f$  são quantidades finitas e discretas, denomina-se esta imagem como de natureza digital. Uma imagem digital é composta por um número finito de elementos, com localização e valor específicos, as quais são geralmente denominados como elementos de imagem ou *pixels* (GONZALEZ; WOODS, 2018).

Figura 6 - Matriz para representação matemática de uma imagem digital

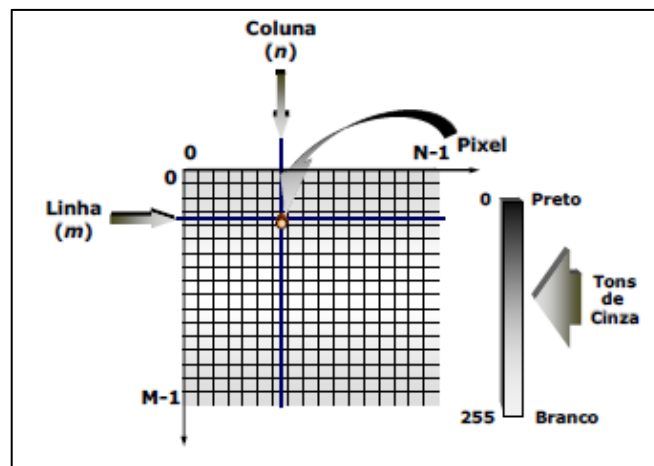
$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix}$$

Fonte: Gonzalez e Woods (2018)

Considera-se que as imagens são baseadas na característica do brilho. Quanto menor este valor, menor será a definição desta imagem para a detecção. Desta maneira, a presença de fontes luminosas deve ser suficiente para que ocorra a reflexão da iluminação nos objetos de cena e a sua visualização não seja impactada.

O valor de amplitude da função  $f(x,y)$  que representa matematicamente uma imagem bidimensional, nas coordenadas espaciais  $(x,y)$ , é uma quantidade escalar positiva cujo o significado físico é determinado pela origem da imagem. Quando uma imagem é gerada a partir de um processo físico, seus valores de intensidade são proporcionais à energia irradiada por uma fonte real (por exemplo, ondas eletromagnéticas). Em consequência,  $f(x,y)$  deve ser diferente de zero e infinito. Esta função pode ser caracterizada por dois componentes: (1) a quantidade de iluminação da fonte que incide na cena que está sendo vista; e (2) a quantidade de iluminação refletida pelos objetos na cena. Esses elementos são denominados respectivamente como componentes de iluminação e refletância (GONZALEZ; WOODS, 2018). A figura 7 demonstra a representação de uma imagem digital bidimensional.

Figura 7 - Representação de uma imagem digital bidimensional



Fonte: Queiroz e Gomes (2006)

As imagens utilizadas em um sistema de processamento digital em geral são capturadas do ambiente por meio de sensores baseados na aquisição de informação a partir de energia irradiada por ondas eletromagnéticas. Dependendo do tipo de sistema, a imagem capturada pode passar por um processo em tempo real ou então pode ser armazenada para que seja utilizada posteriormente.

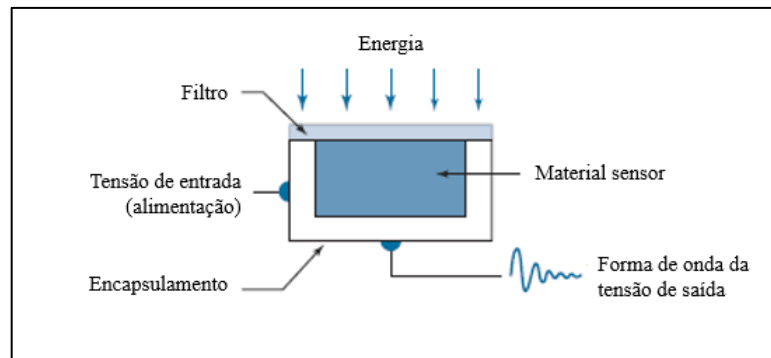
Denomina-se como sequência de vídeo o conjunto de imagens capturadas em sequência (denominados quadros, ou *frames*), dentro de um intervalo de tempo específico, transmitindo ao visualizador a sensação de movimento. Sistemas genéricos digitais de imagens associadas à captura de vídeo podem realizar este tipo de captura, armazená-las formando um banco de imagens e executar suas operações, imagem por imagem, em um curto período de tempo (dependendo da especificação computacional do sistema), sendo este um cenário que serve como base para o desenvolvimento de sistemas de reconhecimento de objetos em tempo real.

### 1.3.2 Aquisição de Imagem

Um sinal digital é o resultado obtido ao realizar a amostragem, quantização e codificação de sinais analógicos, realizando-se a conversão analógica – digital. No campo de estudo de processamento digital de imagens, os sinais analógicos convertidos referem-se à intensidade de iluminação de uma cena que é captada pelo sistema de aquisição de imagem a partir de seus sensores, resultando em um sinal discreto no tempo e que assume valores finitos.

As imagens são geradas em sua maioria pela combinação de uma fonte de “iluminação” e a reflexão ou absorção de energia dessa fonte pelos elementos de “cena” cujo a imagem está sendo gerada. A aquisição de informação referente aos elementos de cena é feita por meio de sensores, conforme ilustra a figura 8. A energia coletada por estes sensores é transformada em tensão pela combinação da energia elétrica de entrada e do material do sensor, sensível a um tipo específico de energia que está sendo detectado. A forma de onda da tensão de saída é a resposta do sensor, e uma quantidade digital é obtida de cada sensor por meio da digitalização de sua resposta (GONZALEZ; WOODS, 2018).

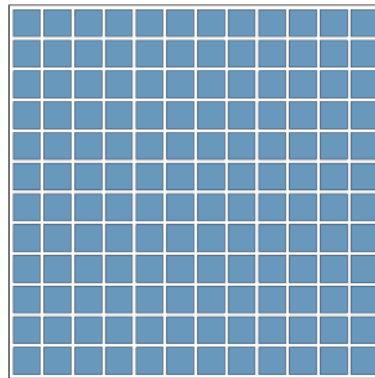
Figura 8 - Sensor de aquisição de imagens (unidade)



Fonte: Gonzalez e Woods (2018)

Um sistema de aquisição de imagens possui um conjunto de matriz de sensores (figura 9) onde ao fazer a captura de uma cena, realiza uma varredura uma por uma, onde cada informação detectada pelos sensores é convertida em níveis de tensão analógica. Os sensores presentes neste sistema detectam a característica de brilho dos objetos presentes na cena e os convertem em níveis de tensão, onde os valores mais aproximados de 0 V representam a ausência de brilho, enquanto que o contrário (acrescimento no valor de tensão), demonstra a presença ou aumento do nível de brilho em uma imagem.

Figura 9 - Sensores de área (matricial)

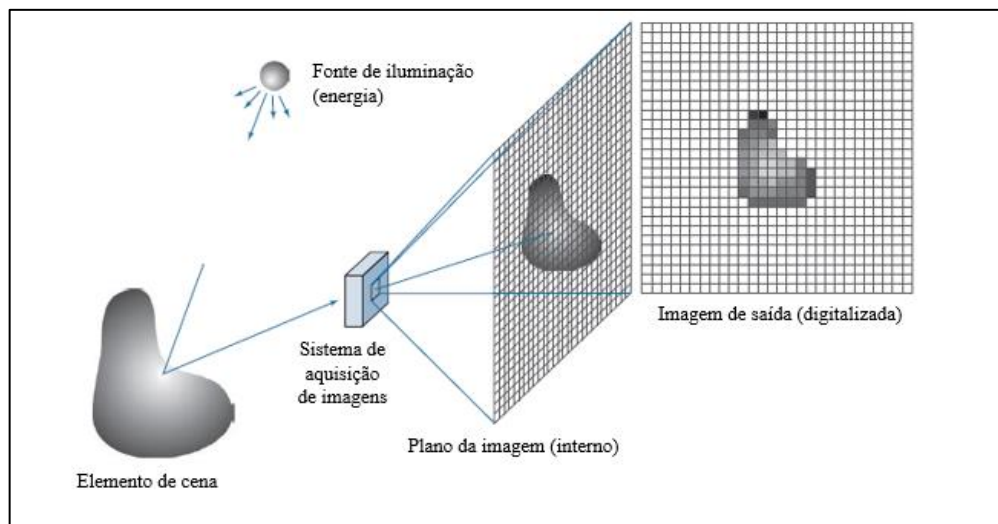


Fonte: Gonzalez e Woods (2018)

Um dos dispositivos de aquisição de imagens mais utilizados é a CCD (*Charged Coupled Device*), comumente utilizada em câmeras digitais, ao qual consiste de uma matriz de células semicondutoras fotossensíveis, que atuam como capacitores, armazenando carga elétrica proporcional à energia luminosa incidente. O sinal elétrico produzido é condicionado por circuitos eletrônicos especializados, produzindo à saída um Sinal Composto de Vídeo (SVC) analógico e monocromático. (MARQUES; VIEIRA, 1999).

A digitalização da imagem ocorre logo após o processo de captura feita pela matriz de sensores do sistema de aquisição de imagens. A informação de brilho convertida em níveis de tensão analógica é digitalizada, ou seja, é realizado um processamento digital do sinal de tensão obtido (amostragem, filtragem e quantização). A informação digitalizada pode ser transferida para um componente de *hardware*, onde podem ser realizadas operações de processamento de imagem, dependendo do tipo de aplicação. O resultado obtido pode ser visualizado em um monitor de vídeo ao fim da etapa de processamento. A figura 10 apresenta uma visão geral de como ocorre o processo de aquisição de uma imagem digital.

Figura 10 - Processo de aquisição de uma imagem digital



Fonte: Gonzalez e Woods (2018)

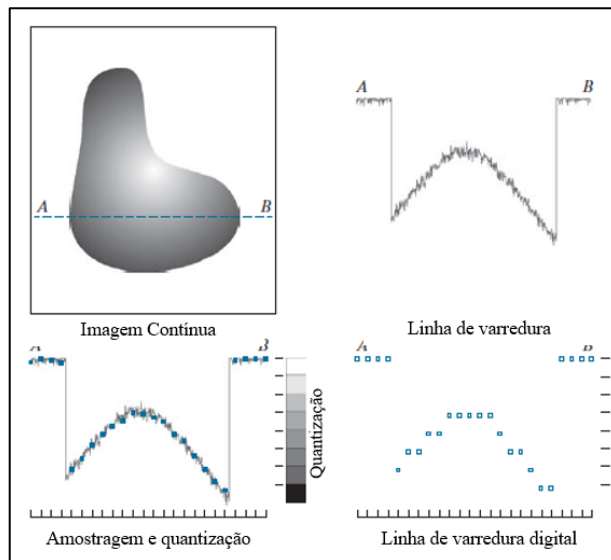
### 1.3.3 Amostragem e Quantização

Os sinais analógicos de tensão provenientes dos sensores do sistema de aquisição de imagem passam por um processamento digital, especificamente pelos processos de amostragem e quantização. No primeiro processo, os sinais de tensão de natureza contínua resultantes da detecção dos sensores serão discretizados, ou seja, o sistema de conversão analógico-digital irá realizar a coleta do valor de amplitude em cada coordenada deste sinal à cada intervalo de tempo baseado na frequência de amostragem ( $f_s$ ) do conversor digital.

A etapa de quantização é caracterizada pela digitalização da amplitude do sinal de tensão de natureza contínua, onde é feita a atribuição do valor de amplitude capturado após amostragem ao seu equivalente em formato de código binário. No caso da digitalização de uma imagem, é atribuído um valor de intervalo de amplitude para cada código pré-estabelecido no

sistema, logo quanto maior o número de combinações, maior será a variedade de níveis de intensidade que podem ser representadas nesta imagem. Em uma imagem em escala de cinzas que utiliza um sistema de 8 *bits*, por exemplo, atribui-se a menor combinação binária à cor preta (equivalente à 0), enquanto que a cor branca recebe o maior código binário (equivalente à 255). As diferentes tonalidades de cinza são distribuídas para as demais combinações que são intermediárias a estes dois extremos. A figura 11 demonstra as etapas de quantização e amostragem de uma imagem, onde são coletadas as informações de intensidade de uma imagem contínua e a partir disso, realiza-se a amostragem da amplitude deste sinal em um intervalo de tempo pré-definido, atribuindo-se um código binário a cada nível de intensidade de acordo com seu valor de amplitude.

Figura 11 - Etapas para produção de uma imagem digital

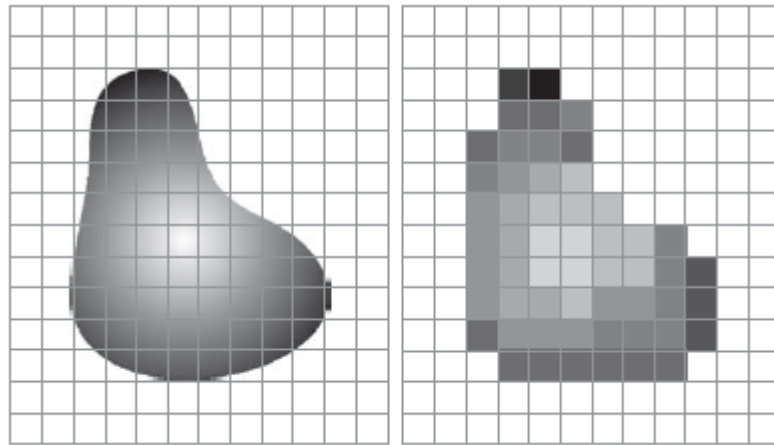


Fonte: Gonzalez e Woods (2018)

Além do número discreto de níveis utilizados para digitalizar a amplitude do sinal, a precisão atingida na quantização depende do conteúdo de ruído do sinal de amostragem, fator este que pode ocasionar falhas, possibilitando a atribuição de um valor de amplitude incoerente ao nível de intensidade dos objetos de cena. A qualidade de uma imagem digital é determinada na maioria dos casos pela quantidade de amostras e de níveis discretos de intensidade utilizados na amostragem e na quantização (GONZALEZ; WOODS, 2018).

A figura 12 demonstra um comparativo entre uma imagem contínua e o resultado equivalente à sua digitalização.

Figura 12 - Imagem contínua projetada em uma matriz de sensores (esquerda) em comparação ao resultado da digitalização da imagem após amostragem e quantização (direita)



Fonte: Gonzalez e Woods (2018)

#### 1.3.4 Imagens Monocromáticas e Policromáticas

Sabe-se que uma imagem digital bidimensional é representada por um conjunto de elementos de imagem (*pixels*) localizados em coordenadas  $x$  e  $y$  do plano de imagem, onde estes possuem valores de intensidade específicas, proporcional ao nível de luminosidade capturados pelos sensores do sistema de aquisição.

Os *pixels* de uma imagem digital podem ser representados de forma monocromática ou policromática. Na primeira alternativa, é atribuída à imagem níveis de tonalidades em escala de cinza, enquanto que na segunda opção a imagem é apresentada em cores.

Os níveis de intensidade dos elementos de imagem são valores finitos, estabelecidos dentro de um intervalo determinado pela quantidade de *bits* da imagem digital. Em imagens de 8 *bits*, por exemplo, cada elemento de imagem é representado por uma sequência de valores binários de oito dígitos, havendo 256 níveis diferentes (de 0 a 255) de tonalidades que podem ser assumidas por um *pixel*.

Em uma imagem em escala de cinzas (monocromática), a imagem digital é representada por uma única camada, cujo os elementos de imagem representam os níveis de intensidade de brilho de uma cena. Os *pixels* dessa imagem podem assumir valores finitos de tonalidade de cinza, como também as colorações branca ou preta, caso o elemento de imagem esteja representando o nível máximo de intensidade de brilho ou a ausência de brilho, respectivamente.



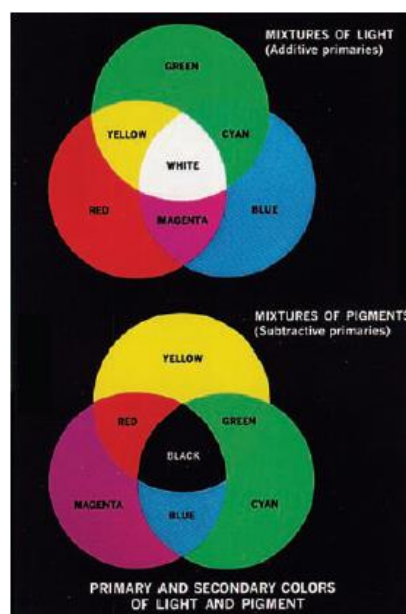
As imagens digitais em cores (policromáticas) são resultantes da combinação de um conjunto de três camadas, onde cada uma carrega uma cor primária, que são: vermelho, verde e azul. A composição destas três cores primárias origina novas tonalidades, logo a representação de uma coloração específica pode ser reproduzida ao combinar estas três cores bastando variar o nível de intensidade de cada cor primária ao ponto de alcançar a coloração desejada. Na figura 13, é apresentada a comparação entre uma imagem colorida e sua versão em escalas de cinzas. Os resultados da combinação das camadas de cores primárias e secundárias são apresentados na figura 14.

Figura 13 - Exemplo de imagem policromática (colorida) e em escala de cinzas



Fonte: Gonzalez e Woods (2018)

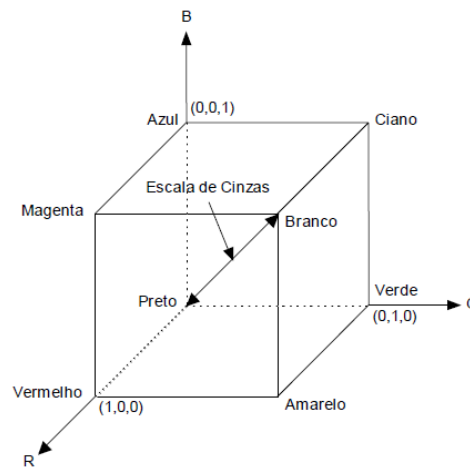
Figura 14 - Cores primárias e secundárias de luz e pigmentos



Fonte: Gonzalez e Woods (2018)

Segundo Marques e Vieira (1999), o modelo mais utilizado para exibição de imagens digitais em cores é o *RGB* (*Red, Green, Blue*), ilustrado na figura 15. Nele a imagem digital é composta por três camadas, uma para cada cor primária: vermelho (R), verde (G) e azul (B). Assim, uma imagem colorida de 8 bits, representada por 3 camadas R, G e B, podem apresentar  $256^3$  (16777216) diferentes tonalidades de cor em cada pixel. Este é o modelo mais utilizado por câmeras e monitores de vídeo.

Figura 15 - Modelo RGB



Fonte: Marques e Vieira (1999)

Além da forma, textura e outras características de uma imagem, a informação de cor é uma característica importante que vêm sendo utilizada exitosamente em várias aplicações de processamento de imagem, tais como reconhecimento de objetos, detecção de imagens equivalentes (*image matching*), recuperação de imagens com base em conteúdo, visão computacional, compressão de imagens coloridas, dentre outros. (ACHARYA; RAY, 2005, tradução nossa).

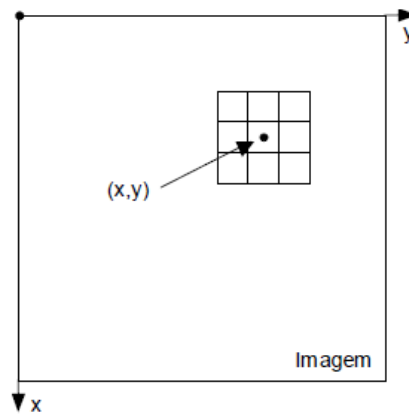
### 1.3.5 Filtragem Espacial

Todos os métodos de processamento de uma imagem que são realizados em seu próprio plano caracterizam-se como técnicas de domínio espacial. As operações matemáticas realizadas dentro desta categoria baseiam-se na manipulação direta de pixels em uma imagem. As duas principais categorias do processamento espacial são as transformações de intensidade e a filtragem espacial.

A filtragem espacial lida com a realização de operações de suavização ou realce de imagens, trabalhando na vizinhança de cada *pixel* de uma imagem. O filtro espacial (também denominado como máscara espacial, *kernel*, *template* ou janela) se trata da aplicação de operações de vizinhança a partir da utilização de uma máscara que está associada à uma operação predefinida a ser aplicada em cada *pixel* de uma imagem de entrada, realizando uma varredura por cada elemento desta imagem. O resultado obtido depende do tipo de operação associada à máscara que foi utilizada (GONZALEZ; WOODS, 2018).

A figura 16 ilustra uma máscara do tipo 3x3, aplicada à um elemento de imagem de coordenadas (x,y).

Figura 16 - Vizinhança/Máscara 3x3 em um ponto de coordenadas (x,y) de uma imagem



Fonte: Marques e Vieira (1999)

Segundo Marques e Vieira (1999), a suavização de imagens no domínio espacial baseia-se no uso de máscaras de convolução adequadas para remoção de ruídos ou aplicar o efeito de desfoque em uma imagem, eliminando detalhes que não são de interesses para as etapas subsequentes de processamento. Dentre as técnicas mais conhecidas, estão a filtragem pela média e o filtro de mediana.

O filtro de média é definido pela utilização de máscaras de convolução compostos por valores unitários, onde a saída é o cálculo da média entre resultado da convolução pela quantidade de valores unitários que estão contidas na região da máscara utilizada para filtragem. Este resultado é substituído ao valor do elemento de imagem cujo foi aplicada esta operação, repetindo-se o mesmo processo para os demais pixels da imagem. O resultado esperado é a perda de nitidez desta imagem e suavização de bordas, que pode ser ampliada com base no aumento de elementos unitários utilizados na máscara. Dentre suas principais aplicações, estão

a redução de ruído de uma imagem, redução de detalhes irrelevantes e suavização de falsos contornos. A figura 17 apresenta os tipos de máscaras que podem ser utilizados para aplicar o filtro de média em uma imagem. A quantidade de elementos interfere no efeito desta operação, neste caso, quanto maior o número de elementos, maior será o efeito de desfoque. A figura 18 ilustra o efeito de um filtro de média quando aplicado em uma imagem.

Figura 17 - Máscaras para cálculo do filtro de média: a) 3x3 , b) 5x5, c) 7x7

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

(a)

$$\frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

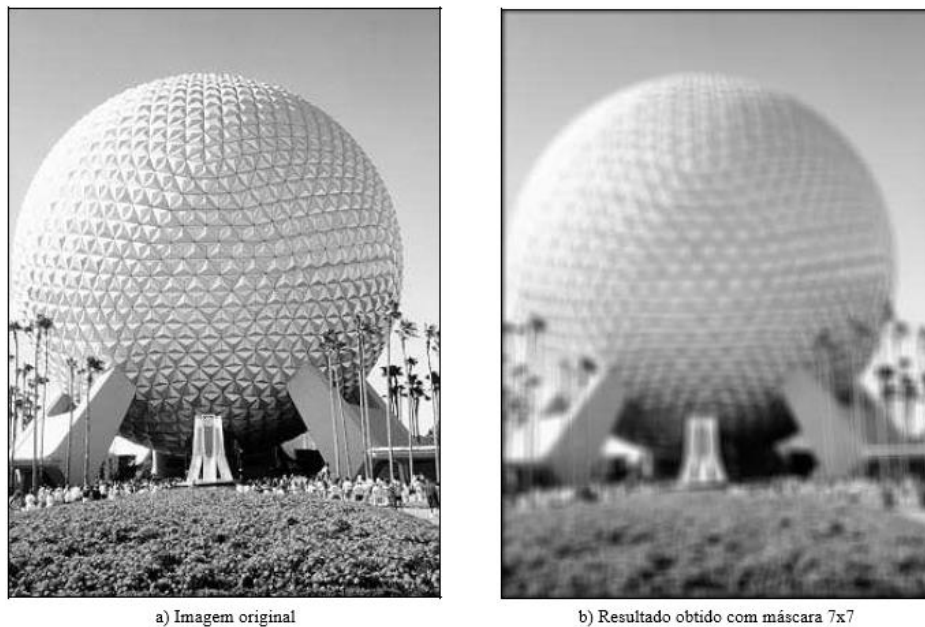
(b)

$$\frac{1}{49} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

(c)

Fonte: Marques e Vieira (1999)

Figura 18 - Utilização do filtro de média para efeito de desfoque



Fonte: Marques e Vieira (1999)

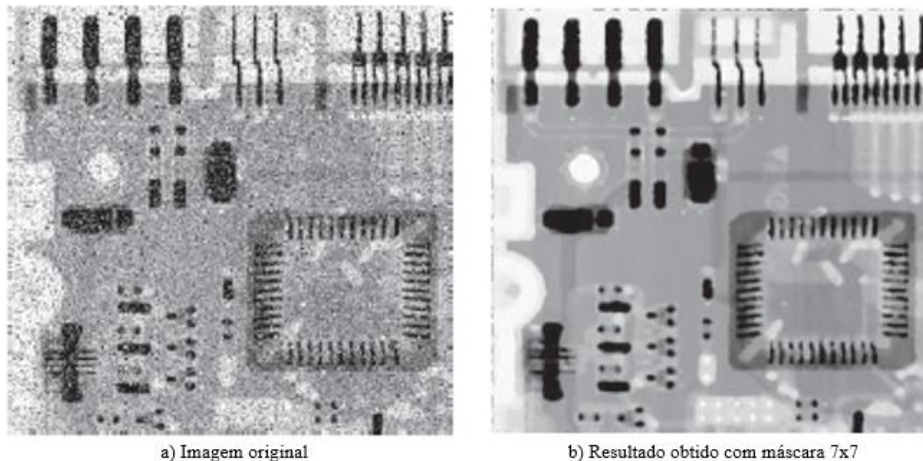
Na técnica de filtro de mediana são coletados todos os elementos de imagem que estão na região delimitada pelas dimensões da máscara utilizada. Todos os elementos são ordenados

com base nos seus valores e a partir disso é feito o cálculo de mediana. Quando o conjunto ordenado é ímpar, a mediana é o valor central deste conjunto. No caso em que a quantidade de elementos ordenados for par, logo o valor de mediana é a média aritmética dos dois elementos mais próximos do centro. O valor a ser atribuído no *pixel* da imagem de saída é o resultado desta mediana. Assim, este é um tipo de técnica que busca atribuir na saída valores dos pixels mais frequentes em torno do elemento de imagem central em que está sendo aplicada a operação, por isso é considerada como uma técnica eficiente para redução de ruído impulsivo (sal-e-pimenta), conforme ilustrado na figura 19.

Considerando uma máscara de dimensões  $n \times n$ , o filtro de mediana tem como saída  $g(x,y)$  o valor de intensidade central do conjunto de pixels que estão presentes na máscara, como apresentado na equação 1:

$$g(x, y) = Md\{f(x_0, y_0), f(x_0, y_1), \dots, f(x_n, y_n)\} \quad (1)$$

Figura 19 - Utilização do filtro de mediana para remoção de ruído (sal-e-pimenta)



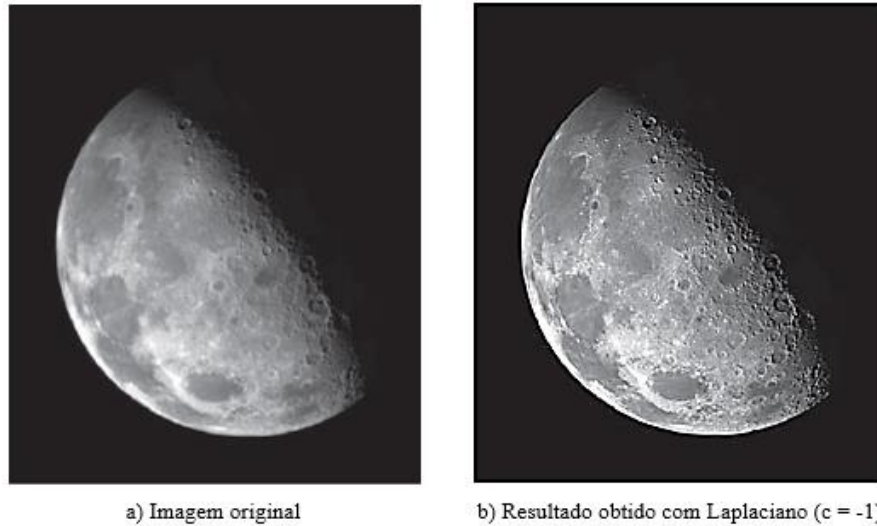
Fonte: Gonzalez e Woods (2018)

As operações de realce (ou aguçamento) atuam em regiões da imagem com componentes de alta frequência, ou seja, locais em que há uma transição de intensidade de *pixels*. Esses tipos de filtros atuam elevando a intensidade dessas regiões, obtendo como resultado a melhoria de nitidez da imagem, conforme ilustra a figura 20.

Os filtros de realce são baseados matematicamente na diferenciação e utilização de derivadas. Fundamentalmente, a força da resposta de um operador derivativo é proporcional ao nível de descontinuidade da intensidade da imagem no ponto no qual o operador é aplicado. Dessa forma, a diferenciação de uma imagem realça as bordas e outras descontinuidades (como

ruído) e atenua as áreas com intensidades de variação mais suave. A operação de realce pode ser realizada de forma mais eficiente e simples a partir de operações de segunda derivada, obtendo-se melhores resultados de aguçamento em comparação com a utilização de operações de primeira derivada (GONZALEZ; WOODS, 2018).

Figura 20 - Utilização do filtro de realce para melhoria de nitidez



Fonte: Gonzalez e Woods (2018)

Segundo Gonzalez e Woods (2018), a utilização do Laplaciano é demonstrada como o operador derivativo isotrópico mais simples para a construção de uma fórmula discreta da derivada de segunda ordem e obtenção de uma máscara para implementação de um filtro de realce. A forma mais básica na qual o laplaciano é utilizado para realce de imagens é apresentado na equação 2.

$$g(x, y) = f(x, y) + c[\nabla^2 f(x, y)] \quad (2)$$

Onde  $f(x,y)$  e  $g(x,y)$  são respectivamente a imagem de entrada e a imagem resultante após o realce. A constante  $c$  pode assumir os valores  $-1$  ou  $1$ , determinando o tipo de máscara de filtragem de realce baseados em laplaciano a ser utilizado.

Na figura 21 são exemplificados quatro tipos de máscaras que implementam o filtro de realce por laplaciano.



Figura 21 - Máscaras para implementação de realce por Laplaciano

0	1	0	1	1	1	0	-1	0	-1	-1	-1
1	-4	1	1	-8	1	-1	4	-1	-1	8	-1
0	1	0	1	1	1	0	-1	0	-1	-1	-1

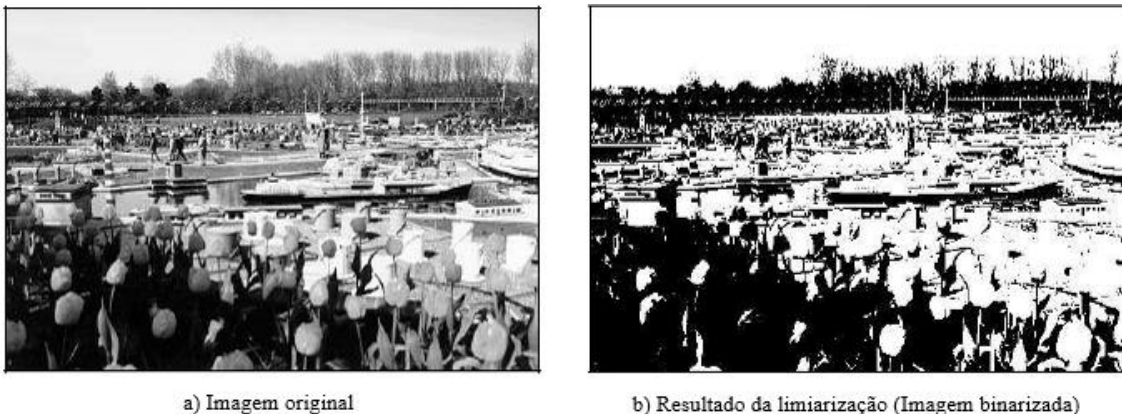
Fonte: Gonzalez e Woods (2018)

### 1.3.6 Limiarização

A limiarização é um método utilizado para realizar o destaque de elementos de imagem que possuem valor de intensidade acima de um limiar pré-estabelecido, atribuindo valores máximos de intensidade, enquanto os demais elementos abaixo desse limite são anulados.

Segundo Marques e Vieira (1999), o princípio da limiarização consiste em separar regiões de uma imagem quando esta apresenta duas classes (o fundo e o objeto). O processo resulta na binarização da imagem, atribuindo ao objeto valores de intensidade máxima, enquanto que o fundo da imagem passa a ter os pixels iguais a zero. A forma mais simples de limiarização consiste na bipartição do histograma, convertendo os elementos de imagem cujo o tom de cinza é maior ou igual a um certo valor de limiar (T) em brancos e os demais em pretos. A figura 22 ilustra essa operação, neste caso foi aplicado um limiar à imagem original onde observa-se como resultado que os valores de intensidade mais altos foram destacados em branco na imagem resultante, enquanto que *pixels* mais escuros foram representados por preto.

Figura 22 - Efeito da utilização de um limiar para binarização de uma imagem



Fonte: Marques e Vieira (1999)

A operação de limiarização de uma imagem pode ser expressa matematicamente descrevendo-a como uma operação de processamento de imagens na qual uma entrada  $f(x,y)$  com um total de  $N$  níveis de cinza produz como saída uma imagem  $g(x,y)$ , chamada imagem limiarizada, em que a quantidade resultante de níveis de cinza é menor do que o total de  $N$  níveis constatados inicialmente na imagem original (MARQUES; VIEIRA, 1999).

A imagem limiarizada  $g(x,y)$  apresenta dois níveis de intensidade, ao qual o valor de saída é determinado seguindo o conceito da equação 3:

$$g(x,y) = \begin{cases} 1, & \text{se } f(x,y) \geq T \\ 0, & \text{se } f(x,y) < T \end{cases} \quad (3)$$

Neste caso, os valores de elementos de imagem igualados à 1 se tratam dos níveis de cinza do objeto, com valores de intensidade de cinza acima do valor limiar ( $T$ ). Para esse caso, é atribuído a estes pixels o nível de intensidade máxima (branco) na imagem de saída. Enquanto que os elementos igualados a 0 correspondem ao fundo (*background*), referentes aos valores abaixo do limiar. Para esses elementos o nível de intensidade de cinza atribuído na saída é mínimo (preto).

### 1.3.7 Morfologia

A morfologia matemática é o estudo que visa a obtenção da estrutura geométrica de entidades presentes em uma imagem. Ela pode ser aplicada em várias áreas de processamento de imagens, como realce, filtragem segmentação, detecção de bordas, afinamento, dentre outras. O princípio básico da morfologia consiste em extrair informações relativas à geometria e à topologia de um conjunto desconhecido (uma imagem), pela transformação através de outro conjunto completamente definido, chamado elemento estruturante (ES). A base da morfologia matemática é a teoria dos conjuntos (MARQUES; VIEIRA, 1999).

A teoria dos conjuntos é amplamente utilizada para o estudo de morfologia matemática. Sua aplicação na área de processamento digital de imagens consiste na utilização de dois conjuntos, onde o primeiro é formado por um objeto da imagem, enquanto o segundo representa um elemento estruturante. A relação entre os dois resulta em um novo conjunto que atribui uma mudança ao objeto da imagem de entrada, porém mantendo o seu formato original.

Os conceitos de reflexão e de translação dos conjuntos são amplamente utilizados na morfologia para formular operações baseadas nos elementos estruturantes, cujo são pequenos



conjuntos ou subimagens usadas para examinar uma imagem buscando propriedades de interesse (GONZALEZ; WOODS, 2018).

A translação de um conjunto  $B$  no ponto  $x = (x_1, x_2)$ , indicada por  $(B)_x$ , é resultante da soma das coordenadas originais dos elementos do conjunto  $B$  com o ponto  $x$ , sendo definida pela equação 4:

$$(B)_x = \{c \mid c = b + x, \quad \text{para } b \in B\} \quad (4)$$

A reflexão de um conjunto  $B$ , indicada por  $\hat{B}$ , é formada pelos elementos do conjunto  $B$ , onde suas coordenadas são opostas ao do conjunto original. Assim, é definida pela equação 5:

$$\hat{B} = \{w \mid w = -b, \quad \text{para } b \in B\} \quad (5)$$

A operação morfológica ocorre realizando-se uma varredura no conjunto formado pelos elementos de um objeto de uma imagem a partir do conjunto do elemento estruturante, que será posicionado em cada um dos pixels do objeto. Dependendo do tipo de operação, a relação entre elemento estruturante e *pixel* irá definir se serão adicionados ou removidos elementos do objeto original, obtendo-se uma imagem ampliada ou comprimida, comparando-se com a imagem de entrada. O comportamento descrito é característico das duas operações fundamentais do processo morfológico: a dilatação e a erosão. Estas são consideradas operações primitivas e que servem como base para diversas opções de algoritmos morfológicos.

#### 1.3.7.1 Dilatação

A operação de dilatação realiza a expansão dos objetos em uma imagem binária, tornando-os mais extensos quando comparados ao seu formato original. O processo básico ocorre a partir do deslocamento do elemento estruturante sobre o conjunto formado pelo objeto da imagem, de maneira análoga à convolução espacial, havendo a expansão nas regiões em que ocorre a interseção entre o conjunto do objeto e do elemento estruturante. O resultado final da expansão da imagem de saída é determinado pelas dimensões e formato do elemento estruturante.

Considerando os conjuntos A e B pertencentes ao conjunto  $Z^2$ , onde A é o conjunto formado pelos elementos do objeto de imagem e B o conjunto composto pelo elemento estruturante, a dilatação de A por B é definida como:

$$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \emptyset\} \quad (6)$$

A equação anterior baseia-se na reflexão de B em torno de sua origem, seguida da translação dessa reflexão por z. A dilatação de A por B é, então, o conjunto de todos os deslocamentos, z, de tal maneira que A e B se sobreponham por pelo menos um elemento (GONZALEZ; WOODS, 2018).

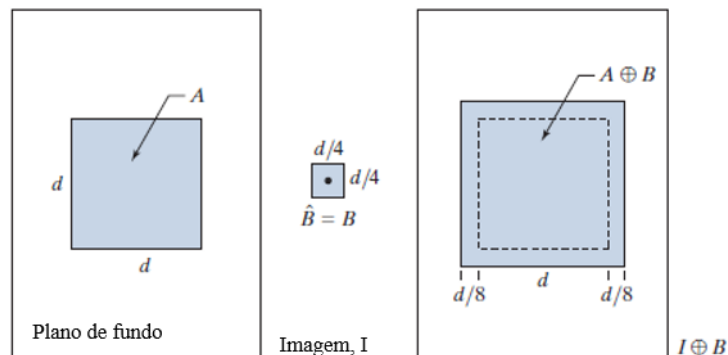
A dilatação de A por B também pode ser descrita de forma equivalente pela equação 7:

$$A \oplus B = \{z \mid [(\hat{B})_z \cap A] \subseteq A\} \quad (7)$$

A utilização do processo de dilatação é considerada útil para o preenchimento de lacunas em objetos de uma imagem. Por exemplo, em uma imagem com veículos em uma via urbana, ao realizar a limiarização e as demais etapas do processamento digital da imagem, os objetos detectados (carros) podem sofrer com a presença de lacunas causadas por sombreamento no ambiente. A partir da operação de dilatação, tais lacunas podem ser corrigidas e o objeto de imagem referente aos veículos recupera aproximadamente o seu formato original.

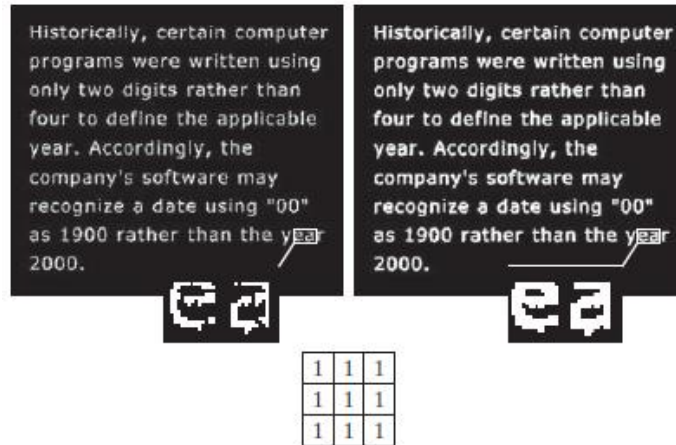
A figura 23 apresenta o processo e os resultados da dilatação de uma imagem. Observe-se que ocorre uma expansão da imagem no mesmo formato do elemento estruturante, nas regiões em que houve a sobreposição de pelo menos um componente deste elemento na imagem.

Figura 23 - Representação do processo de dilatação de uma imagem



A figura 24 apresenta os efeitos da dilatação utilizando um elemento estruturante 3x3 aplicado em um texto. Nota-se que é atribuída à fonte um efeito de destaque após esta operação.

Figura 24 - Representação do efeito da operação de dilatação



Fonte: Gonzalez e Woods (2018)

### 1.3.7.2 Erosão

A operação de erosão tem por objetivo realizar a compressão do formato dos objetos em uma imagem binária, tornando-os menores ou mais finos em relação à imagem original de entrada. O processo básico ocorre de forma similar ao da dilatação, onde é realizado o deslocamento do elemento estruturante sobre o conjunto formado pelo objeto da imagem. A redução da imagem ocorre quando é identificado que o elemento estruturante não está totalmente contido no conjunto formado pelo objeto de imagem. O resultado final da redução é determinado pelas dimensões e formato do elemento estruturante.

Considerando os conjuntos A e B pertencentes ao conjunto  $Z^2$ , onde A é o conjunto formado pelos elementos do objeto de imagem e B o conjunto composto pelo elemento estruturante, a erosão de A por B é definida como:

$$A \ominus B = \{z \mid (B)_z \subseteq A\} \quad (8)$$

Essa equação indica que a erosão de A por B é o conjunto de todos os pontos z de forma que B, transladado por z, está contido em A. Segundo Gonzalez e Woods (2018), como a afirmação de que B (elemento estruturante) deve estar contido em A é equivalente a dizer que

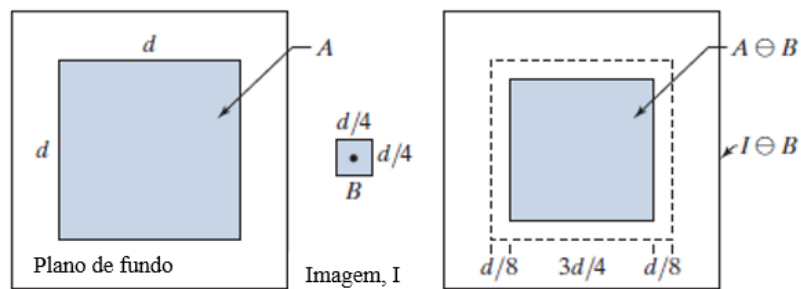
B não tem elementos comuns com o fundo, sendo possível expressar a operação de erosão conforme demonstra a equação 9:

$$A \ominus B = \{z \mid (B)_z \cap A^c = \emptyset\} \tag{9}$$

Onde  $A^c$  é o complemento de A.

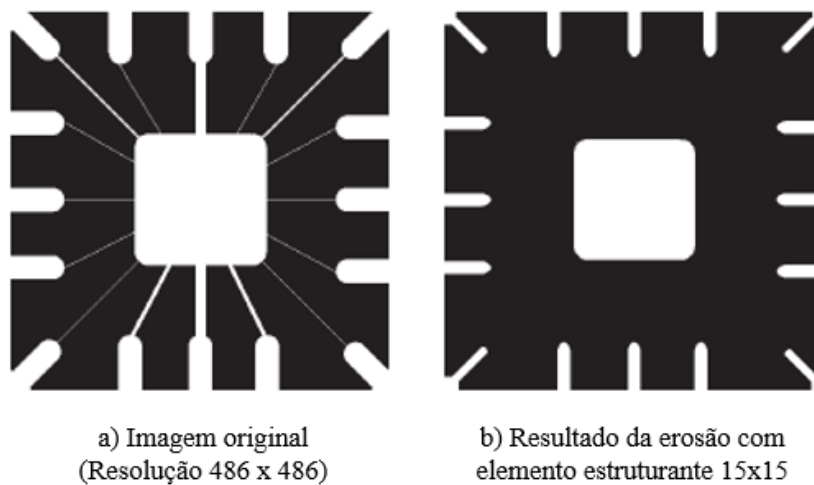
A figura 25 apresenta o processo e os resultados da erosão de uma imagem. Observa-se que ocorre uma compressão da imagem no mesmo formato do elemento estruturante, nas regiões em que pelo menos um componente deste elemento na imagem este sobreposto ao plano de fundo. Os resultados do processo de erosão são representados na figura 26, onde é possível notar que os conjuntos dos elementos de imagem de primeiro plano com área menor ao elemento estruturante foram removidos.

Figura 25 - Representação do processo de dilatação de uma imagem



Fonte: Gonzalez e Woods (2018)

Figura 26 - Representação do efeito da operação de erosão



Fonte: Gonzalez e Woods (2018)

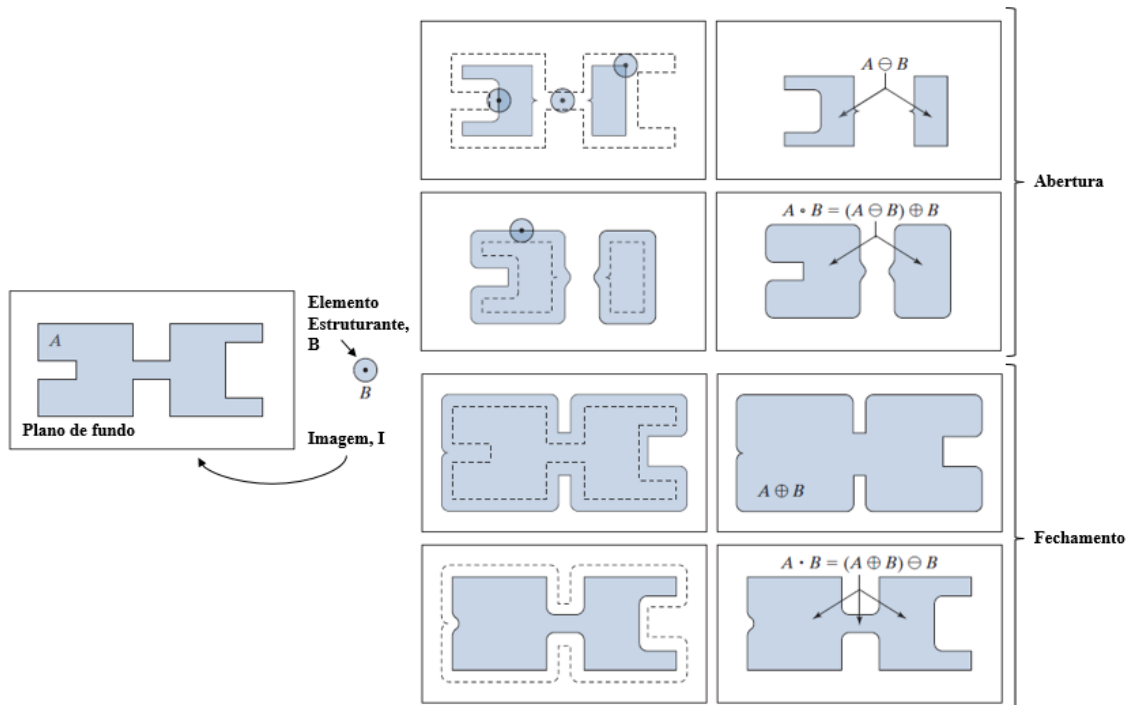
### 1.3.7.3 Abertura e fechamento

A abertura e fechamento são operações morfológicas que resultam na suavização de contornos de objetos quando aplicadas em uma imagem digital. Estas operações também servem para o preenchimento de lacunas que podem surgir ao realizar o processamento de um objeto de imagem e realizar o tratamento de discontinuidades. Ambos os processos são baseados na combinação das operações de dilatação e erosão, sendo consideradas úteis para a construção de filtros morfológicos.

A abertura é a operação morfológica que geralmente suaviza o contorno de um objeto, rompe os istmos e elimina as saliências finas. O fechamento também tende a suavizar contornos, mas, ao contrário da abertura, geralmente funde as discontinuidades estreitas e alonga os golfos finos, elimina pequenos buracos e preenche as lacunas em um contorno (GONZALEZ; WOODS, 2018).

A figura 27 ilustra a execução das operações de abertura e fechamento, aplicados separadamente à uma mesma imagem.

Figura 27 - Representação das operações morfológicas de abertura e fechadura



Fonte: Gonzalez e Woods (2018)

A abertura do conjunto A pelo elemento estruturante B ( $A \circ B$ ), é obtida a partir da erosão de A por B, seguida da dilatação do resultado por B, logo:

$$A \circ B = (A \ominus B) \oplus B \quad (10)$$

O fechamento do conjunto A pelo elemento estruturante B ( $A \cdot B$ ), é obtido a partir da dilatação de A por B, seguido da erosão do resultado por B, sendo definida de acordo com a equação 11:

$$A \cdot B = (A \oplus B) \ominus B \quad (11)$$

#### 1.4 VISÃO COMPUTACIONAL

A visão computacional é a integração e a automatização de uma ampla gama de processos e representações utilizadas para percepção visual. Nela estão incluídos conceitos técnicos de processamento de imagens (transformação, codificação e transmissão de imagens), classificação estatística de padrões (teoria de decisão estatística aplicada a padrões gerais, visuais e dentre outros), modelagem geométrica e processamento cognitivo. A visão computacional é a ciência que estuda e desenvolve tecnologias que tornam possível que sistemas ou máquinas tenham a capacidade de reconhecer características, padrões e objetos do meio, através de imagens capturadas a partir de sensores e dispositivos de captura de imagens. Os dados coletados, quando processados, permitem realizar tarefas, como por exemplo o reconhecimento de objetos. (BALLARD, 1982, p. 2, tradução nossa).

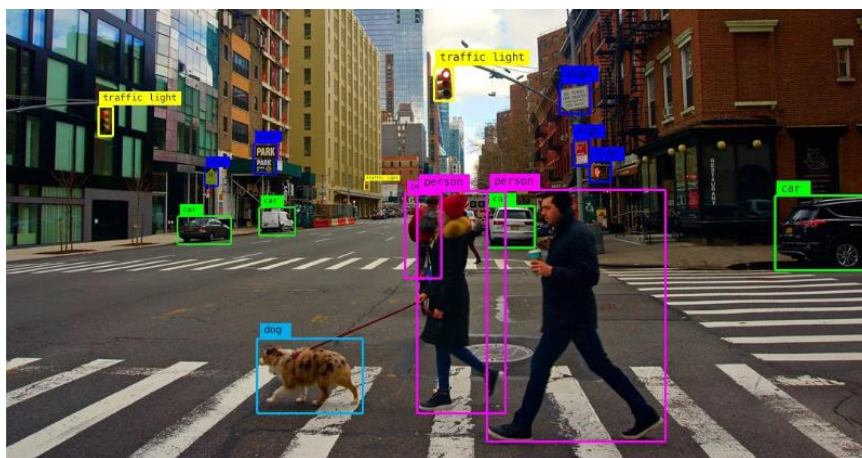
As técnicas de visão computacional são consideradas eficientes para a identificação e reconhecimento de objetos, possibilitando a aquisição desses tipos de dados com uma acurácia superior em relação a outros tipos de ferramentas de sensoriamento, além de ter a capacidade de classificar o tipo de informação detectado de maneira eficiente quando comparado com outros tipos de métodos adequados para esta função.

“O reconhecimento de objetos é uma das principais funções da área de visão computacional e está relacionado diretamente com o reconhecimento de padrões. Um objeto pode ser definido por mais de um padrão (textura, forma, cor, dimensões, etc.) e o reconhecimento individual de cada um destes padrões podem ser divididas em dois grandes grupos: estruturais, onde os padrões são descritos de forma simbólica e a estrutura é a forma como estes padrões se relacionam; o outro grupo é baseado em técnicas que utilizam teoria de

decisão, neste grupo os padrões são descritos por propriedades quantitativas e deve-se decidir se o objeto possui ou não estas propriedades.” (MARENGONI et al, 2010, p.20).

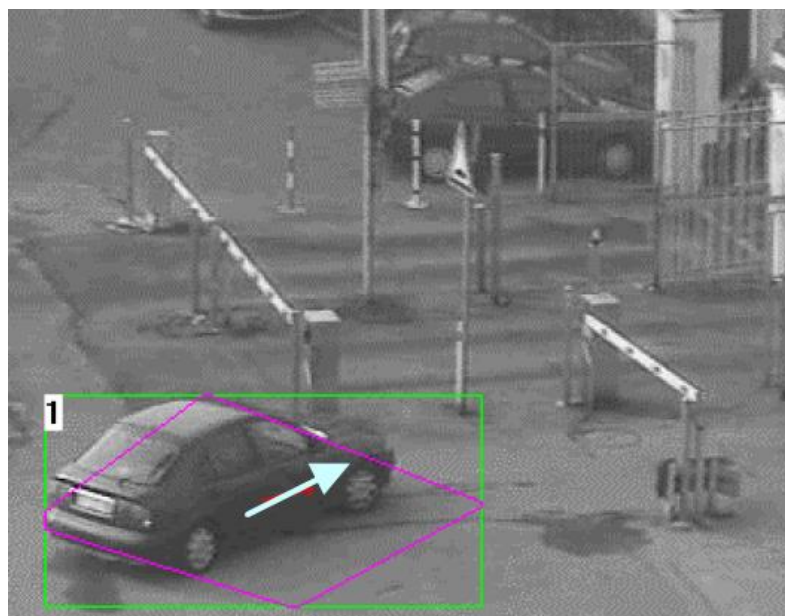
A detecção de objetos permite que sistemas identifiquem elementos de um cenário, estejam eles em movimento ou não, assim como classificá-los, conforme ilustrado na Figura 28 e 29. Para que ocorra esta detecção, o sistema de aquisição de imagens deve utilizar-se de abordagens de processamento digital de imagens e inteligência artificial, sendo estas as principais áreas que dão origem à visão computacional.

Figura 28 - Reconhecimento de objetos a partir de visão computacional



Fonte: Algotive (2023)

Figura 29 - Exemplo de aplicação de visão computacional para detecção veicular



Fonte: Cucchiara et al (2000)

A detecção de objetos pode ser implementada a partir de diversas abordagens, dentre as quais podem ser destacadas:

- a) *Background Subtraction Method (BSM)*;
- b) *Haar Cascades*;
- c) *Histogram of Oriented Gradients (HOG)*;
- d) Redes Neurais Convolucionais;
- e) R-CNN;
- f) *Fast R-CNN*;
- g) *Faster R-CNN*.
- h) YOLO (*You Only Look Once*).

## 1.5 SUBTRAÇÃO DE FUNDO DE IMAGEM

A Subtração de Fundo (*Background Subtraction*) é um método caracterizado pela remoção do plano de fundo de uma imagem, mantendo-se apenas os objetos (primeiro plano) na imagem de saída, ou seja, resulta em uma imagem binária onde são mantidos os *pixels* dos objetos que são novos no cenário ou que estão em movimento na cena. O processo é muitas vezes utilizado em bancos de imagens ou vídeos obtidos por uma câmera em posição fixa de forma que o plano de fundo apresenta característica estática, sendo possível detectar objetos (ou imagem de primeiro plano) em movimentação quando aplicar-se o método. Na subtração de fundo, basicamente é realizada a diferença entre uma imagem de referência e a imagem de plano de fundo, subtraindo-se cada elemento de imagem, ponto a ponto.

O método de subtração de fundo busca qualquer tipo objetos da imagem que não apresentem as características do plano de fundo. Para esse tipo de aplicação, uma segmentação eficiente pode ser obtida muitas vezes subtraindo a aparência estimada do plano de fundo da própria imagem e então buscando por grandes valores absolutos no resultado desta diferença (FORSYTH, 2002 – tradução nossa).

Uma das abordagens mais práticas para realizar a detecção de mudança em dois quadros é a da comparação pixel a pixel entre as duas imagens. Supondo a utilização de uma imagem de referência contendo apenas elementos estacionários, tal como um plano de fundo de característica estática, e diferenciando-a com um quadro posterior no mesmo cenário que contém um novo objeto, o resultado será a eliminação dos elementos fixos e no manutenção dos elementos de imagem cujo a diferença foi maior do que zero, onde estes correspondem aos



componentes não estacionários da imagem, também denominados como objeto (GONZALEZ; WOODS, 2018).

A imagem de diferença entre as duas imagens capturadas nos momentos  $t_i$  e  $t_j$  pode ser definida como:

$$d_{ij}(x, y) = \begin{cases} 1, & \text{se } |f(x, y, t_i) - f(x, y, t_j)| > T \\ 0, & \text{caso contrário} \end{cases} \quad (12)$$

Na qual  $T$  é um limiar estabelecido,  $x$  e  $y$  são coordenadas do plano da imagem.

Para aplicações de subtração de fundo de imagem, a diferença é realizada subtraindo-se uma imagem de referência  $I_t(x, y)$  com a imagem definida como modelo de plano de fundo  $B_t(x, y)$ . Assim, baseando-se na equação 12, a imagem resultante  $V_t(x, y)$  desta operação é dada por:

$$V_t(x, y) = \begin{cases} 1, & \text{se } |I_t(x, y) - B_t(x, y)| > T \\ 0, & \text{caso contrário} \end{cases} \quad (13)$$

Nota-se que a partir relação apresentada e com base nos conceitos do método de subtração de fundo, um objeto é detectado sempre que a diferença entre dois *frames* subsequentes de um vídeo apresentar em sua localização uma diferença maior que o limiar estabelecido, assim, os objetos são representados na imagem binária resultante por meio dos elementos de nível de intensidade mais alto (branco), conforme ilustra a figura 30. De forma contrária, quando a diferença entre os elementos de imagem dos dois *frames* está abaixo do limiar considera-se que há uma similaridade entre ambas as regiões, concluindo que esta localização se trata da imagem de fundo, atribuindo-se o valor mínimo (preto) nesta região da imagem de saída.

Figura 30 - Diferença entre imagens realizada por subtração de fundo



A prática da subtração de fundo não-adaptativa é considerada como a aplicação mais simples do método, onde é feita uma captura do plano de fundo sem haver a presença de qualquer tipo de objeto, utilizando-a como referência para ser subtraído de cada *frame* atual de vídeo capturado pela câmera. Esta é uma abordagem que pode ser suscetível a falhas de detecção, tendo em vista que o tipo de ambiente pode sofrer por alterações, seja por fatores de iluminação ou por fatores naturais (presença de névoa no local, por exemplo). Todavia, podem ser utilizados algoritmos que permitem que seja feita a modelagem do plano de fundo por meio de métodos estatísticos e probabilísticos, possibilitando a atualização deste plano de fundo de forma adaptativa.

A subtração de fundo tem como componente principal a etapa de modelagem do plano de fundo da imagem. A partir dela, é possível obter um bom desempenho na segmentação assim como possibilitar a melhoria da detecção dentre variações do ambiente, tais como mudança de iluminação. Existem diferentes técnicas que possibilitam a modelagem de um plano de fundo, as quais são definidas em dois tipos, recursivas e não-recursivas. De acordo com Cheung et al (2003), ambas as técnicas podem ser descritas conforme abaixo:

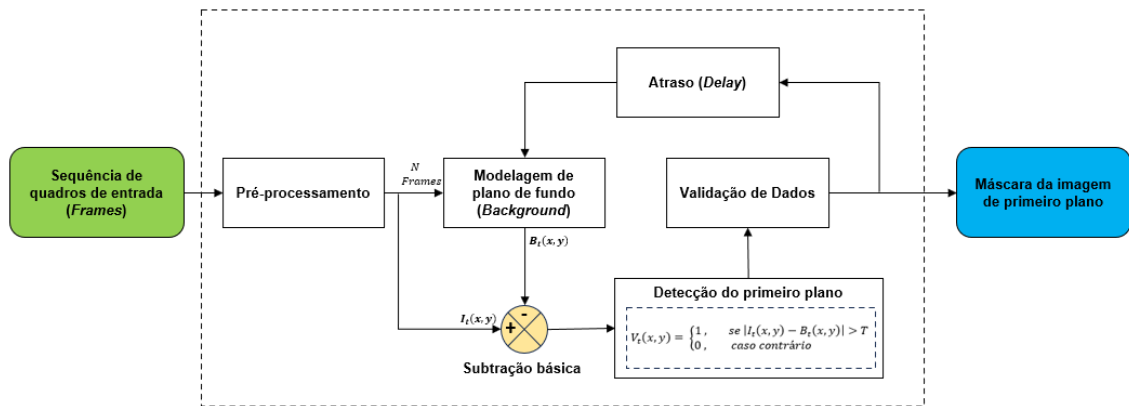
- **Técnicas de modelagem não-recursivas:** São técnicas de modelagem de plano de fundo que utilizam um *buffer* para armazenamento de um conjunto L de *frames* de vídeo anteriores, estimando o *background* com base na variação temporal do nível de intensidade de cada *pixel* amostrado. Estas técnicas têm como característica a alta adaptabilidade tendo em vista que a modelagem está restrita apenas ao intervalo de *frames* prévios amostrados.
- **Técnicas de modelagem recursivas:** As técnicas de modelagem recursiva não utilizam um *buffer* para realizar a estimativa do plano de fundo. Ao invés disso, nesta técnica é feita uma atualização recursiva em um modelo de *background* único à cada entrada de um novo *frame*. São caracterizadas por requisitar menos espaço de armazenamento e atribuir valores de peso ponderado para descontar quadros passados.

Recomenda-se a utilização das técnicas de subtração de fundo em vídeos que possuem característica de ambiente de fundo estático, com imagens capturadas a partir de uma câmera com posicionamento fixo e em um ambiente cujo a iluminação possa ser bem distribuída. De forma geral, o funcionamento de uma técnica de subtração de fundo deve cumprir com as seguintes etapas:

- 1) **Pré-processamento:** é o processamento de um conjunto de *frames* capturados do vídeo de entrada que serão utilizados para realizar a modelagem do plano de fundo e para subtração dos quadros. Nesta etapa, em cada quadro são aplicadas operações fundamentais de processamento de imagens para permitir a implementação das fases subsequentes do método de subtração de fundo.
- 2) **Modelagem de fundo:** é a obtenção de um modelo de fundo da imagem a partir dos *frames* capturados pela câmera de vídeo na etapa de pré-processamento. Em técnicas de modelagem recursivas, as etapas de pré-processamento e modelagem de fundo devem repetir em um intervalo de tempo pré-estabelecido (*delay*), permitindo a constante atualização do modelo de fundo. A modelagem de background é realizada em geral com o apoio de abordagens estatísticas.
- 3) **Subtração básica:** nesta etapa é feita a diferença entre o *frame* atual e o modelo de plano de fundo utilizando o conceito da subtração simples entre dois quadros de vídeo.
- 4) **Detecção do primeiro plano (*foreground*):** é a etapa em que o algoritmo realiza a decisão, por meio de um limiar pré-estabelecido (*threshold*), sobre quais elementos de imagem se tratam do primeiro plano (objeto) e quais pertencem ao plano de fundo, com base no resultado da intensidade de cada *pixel* na etapa de subtração básica. O resultado obtido é uma imagem binarizada onde os objetos são representados pela cor branca e o fundo representado pela cor preta.
- 5) **Validação de dados:** é o momento em que a imagem binarizada é corrigida ou filtrada, removendo-se os pixels que não correspondem aos objetos que estão em movimentação no vídeo, resultando na máscara da imagem de primeiro plano, cujo é utilizada para indicar a localização dos objetos que estão em movimentação no vídeo resultante.

A figura 31 ilustra os componentes anteriormente citados em formato de diagrama, apresentando as etapas em que ocorrem o método de subtração de fundo de uma imagem, desde a imagem de entrada até o resultado do processamento com a imagem binária, também denominada como a máscara obtida por subtração de fundo.

Figura 31 - Diagrama geral de um algoritmo de subtração de fundo



Fonte: Autoria própria

Dentre as vantagens da utilização de subtração de fundo é possível citar a princípio a sua simplicidade de implementação e o seu desempenho em termos processamento, sendo uma abordagem que é caracterizada pela baixa necessidade de poder de computacional.

Abaixo são apresentadas as principais áreas para utilização da abordagem de subtração de fundo:

- Monitoramento de tráfego;
- Rastreamento de objetos;
- Vigilância e segurança eletrônica.

Atualmente existem diversas técnicas para a implementação do método de subtração de fundo. Cada opção possui sua característica e recomendação dependendo do tipo de aplicação. Dentre as opções abordadas neste trabalho, estão técnicas de modelagem GMG, KNN, CNT, MOG e MOG2. Os tópicos seguintes apresentam as descrições gerais de cada técnica em relação aos seus respectivos modos de operação.

### 1.5.1 Método de Godbehare, Matsukawa, Goldberg (GMG)

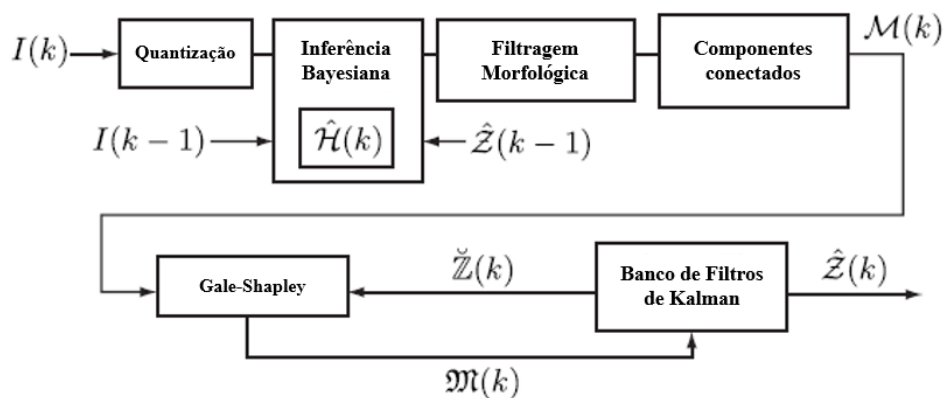
O algoritmo de subtração de fundo GMG, nomeado a partir das iniciais dos autores desta técnica (Andrew Godbehare, Akihiro Matsukawa, Ken Goldberg) e publicado em 2012, apresenta uma abordagem de subtração de fundo recursiva que combina os conceitos probabilísticos e estatísticos do Teorema de Bayes em cada pixel para modelagem do plano de fundo e do Filtro de Kalman para detecção dos possíveis objetos de primeiro plano.

A primeira etapa desta técnica consiste na utilização de um algoritmo probabilístico para a segmentação da imagem de primeiro plano aplicado aos quadros do vídeo. Este primeiro algoritmo irá identificar possíveis objetos da imagem utilizando o conceito de inferência Bayesiana com um modelo estimado de plano de fundo variante no tempo e um modelo de fundo inferido. Ainda nesse momento, são atribuídos valores de peso para cada *pixel* que vão acumulando com o tempo conforme o elemento mantém-se na mesma intensidade ou coloração. Na segunda etapa é feita a filtragem dos elementos de imagem que foram considerados como *pixels* do primeiro plano, com o objetivo de reduzir o ruído decorrente do primeiro estágio do processo. Neste momento é realizado um refinamento e uma filtragem seletiva direcionado aos possíveis objetos da imagem, assim como o rastreamento de objetos, aplicado a partir de um conjunto de filtros de Kalman. Este filtro tem como propósito minimizar o erro quadrático, reduzindo características de ruído, utilizando medições realizadas ao longo do tempo (que podem conter ruído) e gerar resultados que tentam se aproximar aos valores reais das grandezas medidas. Como resultado, o algoritmo permite a atualização seletiva do modelo de fundo da imagem, evitando regiões que foram identificadas anteriormente como primeiro plano (GODBEHERE et al, 2012, tradução nossa).

O algoritmo possui característica recursiva, ao qual o modelo de plano de fundo é atualizado à cada *frame*. O preceito do algoritmo tem como base a característica estática dos níveis de intensidade considerando como plano de fundo os elementos de imagem que mantém o nível de intensidade estável por um longo tempo (GODBEHERE et al, 2012).

A figura 32 ilustra as etapas que compõem a técnica de modelagem GMG.

Figura 32 - Diagrama de blocos do algoritmo GMG



Fonte: Godbehere et al (2007)

O teorema de Bayes descreve a probabilidade da ocorrência de um evento, baseado em um conhecimento anterior que pode ser relacionado ao evento atual a ocorrer. O teorema aplicado no algoritmo GMG considera a probabilidade de um *pixel* ser classificado como um elemento do primeiro plano (F) ou um elemento pertencente ao plano de fundo (B) dada a característica observada  $f_{ij}(k)$ , assim a probabilidade de um elemento pertencer ao background em um instante  $k$  dado o evento ocorrido do elemento  $f_{ij}(k)$ :

$$p(B|f) = \frac{p(f|B)p(B)}{p(f|B)p(B) + p(f|F)p(F)} \quad (14)$$

No algoritmo GMG, o processamento se inicia ao quantizar a imagem de entrada no espaço em cores, comparando-o com o modelo estatístico de imagem de plano de fundo, para gerar uma imagem probabilística posterior. A cada novo *pixel*, o algoritmo realiza a classificação do elemento de acordo com a probabilidade definida pela inferência bayesiana, onde é calculada a probabilidade deste novo elemento de imagem fazer parte ou não do plano de fundo. Essa imagem é filtrada por operações morfológicas de abertura e fechamento e então é segmentada em um conjunto de *bouding boxes* pelos componentes do algoritmo. O banco de filtros de Kalman mantém um conjunto de objetos previamente rastreados para composição do modelo, sendo um método utilizado para a extração de ruído durante o processamento, auxiliando na obtenção de classificações e estimativas mais confiáveis para os elementos de imagem que compõe o objeto detectado (GODBEHERE et al, 2012, tradução nossa).

Uma das vantagens apresentadas pelos autores do método GMG está na melhoria na taxa de detecção de verdadeiros positivos, em comparação a outros algoritmos de subtração de fundo.

Apesar de ser um método recursivo, o algoritmo GMG requer uma quantidade elevada de amostras para que seja feito o treinamento e modelagem do plano de fundo inicial, quando comparado aos demais algoritmos de subtração de fundo. No entanto, é um fator que não implica em uma melhoria significativa em relação à detecção e remoção de ruídos na imagem segmentada (TRNOVSZKÝ et al, 2017).

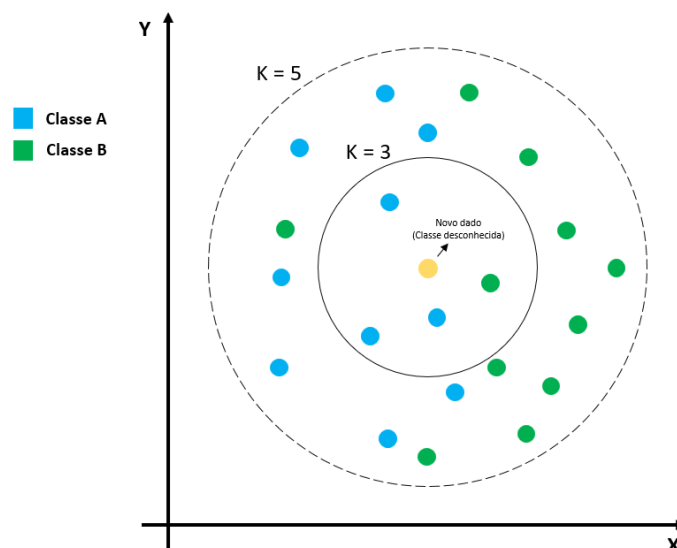
### 1.5.2 *K-nearest Neighbors* (KNN)

O método dos K-vizinhos mais próximos (*K-nearest neighbors* – KNN) utiliza-se da análise de dados que pertencem a classes distintas, sendo representados por pontos em um dado

plano cartesiano. Ao incluir um novo dado ao qual sua classe é desconhecida, o mesmo é representado e incluído como um ponto sem classificação no plano. O método utiliza-se da heurística de que cada ponto não classificado deve ser categorizado e incluído na classe predominante em suas proximidades. Dessa forma, é atribuído ao dado sem categoria a mesma classe pertencente aos pontos que compõem a maioria em sua vizinhança.

Um classificador de vizinhos mais próximos identifica uma quantidade  $k$  de pontos que estão mais próximos do ponto de classe desconhecida. Os pontos que estão dentro da vizinhança são contabilizados de acordo com sua classe e aquela que representa a maioria dentro da vizinhança será considerada como a categoria a ser atribuída ao ponto desconhecido, seguindo o formato conceitual demonstrado na figura 33. Um parâmetro limitante  $l$  pode ser incluído no algoritmo e caso a contabilização das classes não ultrapasse esse limite, conclui-se que o ponto é pertencente à uma classe desconhecida. Para o caso em que o parâmetro  $l = 0$ , a classe a ser atribuída ao ponto deve ser aquela em que houver a maioria em termos de representatividade, ou seja, o ponto é classificado com base nos  $k$ -vizinhos mais próximos (FORSYTH, 2002, tradução nossa).

Figura 33 - Classificação de dados por KNN



Fonte: Autoria própria

Davis e Rosenfeld (1978) apresentam o conceito de vizinhos mais próximos aplicados em imagens em escala de cinzas, onde um ponto  $P$  (novo dado) recebe um novo valor de intensidade de cinza baseado na média dos valores de intensidade dos  $K$  pontos vizinhos de  $P$ , dentro de uma vizinhança  $n \times n$  em torno de  $P$ .

O método KNN aplicado à segmentação do plano de fundo de uma imagem irá considerar cada *pixel* como sendo cada um dos pontos distribuídos no plano, aos quais devem pertencer a duas classes distintas: primeiro plano (objeto) e plano de fundo. Estas são definidas a partir de uma quantidade de amostras previamente capturadas. Ao receber um novo *pixel* neste conjunto de dados, o algoritmo irá calcular a distância euclidiana deste ponto aos elementos de imagem das classes de plano de fundo e primeiro plano.

A equação 15 apresenta o cálculo da distância euclidiana entre dois pontos A e B:

$$d_{AB} = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2} \quad (15)$$

Será calculada a distância euclidiana (equação 15) para cada ponto pertencente às duas classes presentes na imagem. Além disso, é necessário estabelecer um raio de valor K em torno do ponto de categoria desconhecida. Para cada distância contabilizada dentro do raio pré-determinado, é feita a incrementação do contador referente à categoria que está sendo feita a medição em relação ao ponto. A classe a ser atribuída a este novo elemento será aquela que pertence ao grupo em que foi contabilizada a maior quantidade de *pixels* dentro do raio definido.

O conceito base do método KNN pode ser combinado com outras abordagens para aprimoramento da segmentação do plano de fundo. Qasim et al (2021), apresenta a utilização do método combinado ao cálculo de estimativa de densidade.

O método KNN apresenta uma abordagem de subtração de fundo a partir de um conceito simples, no entanto é um algoritmo que é suscetível à ruídos, o que pode comprometer o seu desempenho em comparação a outros tipos de abordagens (MATCZAK; MAZUREK, 2021).

Qasim et. al (2021) recomenda a utilização de técnicas de *machine learning* ou *deep learning* em conjunto com KNN visando a melhoria de desempenho a partir desta abordagem.

### 1.5.3 Count (CNT)

O método CNT (*CouNT*) é considerado como uma das abordagens mais rápidas para implementação da subtração de fundo de uma imagem. O funcionamento do algoritmo em si também é considerado simples, construído a partir de um conjunto de funções da biblioteca OpenCV, este irá realizar a contagem de uma quantidade específica de *frames* anteriores do vídeo, tomando-os como amostras. A partir disso, cada *frame* coletado é comparado entre si, avaliando-se o comportamento do nível de intensidade de cada um dos elementos de imagem no decorrer da variação dos quadros.



O algoritmo identifica a variação de intensidade de cada elemento de imagem durante o percorrer dos quadros amostrados e então realiza uma decisão com base nesse comportamento. Caso o algoritmo identifique que houve uma elevada taxa de variação entre os *frames* no elemento de imagem em sua coordenada específica, considera-se que este elemento compõe o primeiro plano. Caso contrário, se um *pixel* mantém seu nível de intensidade adotando um comportamento estável entre os quadros, este elemento é considerado como um componente do plano de fundo.

São utilizados dois termos para indicar os limites de contagem em um algoritmo CNT e ambos são incluídos como parâmetros de entrada para a função que implementa a técnica. O primeiro parâmetro, é denominado como o limite mínimo para considerar um *pixel* estável, ou seja, a quantidade mínima de *frames* em que um elemento de imagem deve manter-se aproximadamente com o mesmo valor para que este seja considerado como parte do plano de fundo. Caso haja uma variação significativa dentro deste intervalo, o mesmo é definido como primeiro plano. O segundo limite estabelece o valor máximo de *frames* que devem ser contados para que seja iniciada uma nova atualização do plano de fundo.

De acordo com o desenvolvedor do algoritmo, Sagi Zeevi, o CNT apresenta uma maior velocidade em termos de tempo de processamento em comparação a outros tipos de técnicas para subtração de fundo, como por exemplo o método de mistura de gaussianas (MOG). Além disso, o algoritmo CNT é apresentado como uma técnica que possui uma melhoria de performance quando implementado a partir de sistemas com baixas especificações de *hardware*.

#### 1.5.4 Mistura de Gaussianas (MOG/MOG2)

O método de mistura de gaussianas (*Mixture of gaussians* – MoG) é uma das abordagens recursivas mais usuais para a implementação da técnica de subtração de fundo. Esta técnica é baseada no conceito de que as intensidades dos *pixels* da imagem em análise podem ser modeladas a partir de uma combinação de distribuições gaussianas. Este algoritmo realiza o aprendizado do modelo de fundo da imagem, onde são feitos os cálculos para a determinação das distribuições gaussianas de cada *pixel* que serão utilizadas para classificação dos futuros elementos de imagem como plano de fundo ou de primeiro plano.

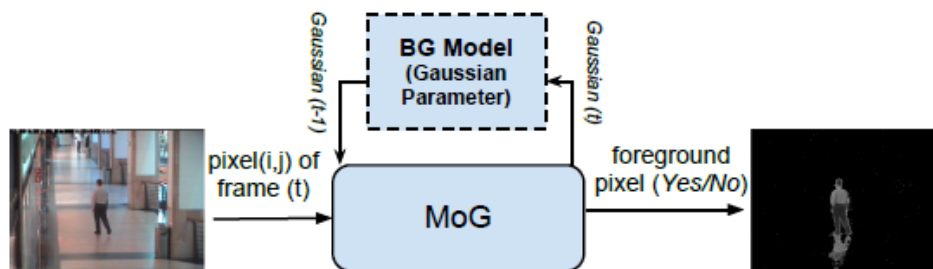
Segundo Acharya e Ray (2005), um *pixel* em particular pertencente a uma imagem pode ser modelado a partir de uma combinação de distribuições gaussianas, considerando que no decorrer de um vídeo, esse mesmo elemento pode assumir diferentes níveis de intensidade

devido a passagem de objetos em sua localização. As distribuições Gaussianas representam o comportamento probabilístico em relação ao nível de intensidade dos *pixels* e com isso podem ser utilizadas para a modelagem do plano de fundo.

A abordagem de subtração de fundo a partir da mistura de gaussianas foi apresentada por Stauffer e Grimson em 1999. Segundo os autores da publicação, este método é baseado na persistência e na variância de distribuições Gaussianas que são referentes à intensidade de um único pixel e representam os intervalos de valores que correspondem à característica de plano de fundo. Dessa forma, ao realizar a leitura de um novo pixel, o seu valor de intensidade deve estar localizado dentro de uma das regiões especificadas pelas distribuições Gaussianas para que seja considerado um pixel pertencente ao plano de fundo. Se o elemento de imagem não se encaixar em nenhuma das distribuições Gaussianas, logo o mesmo deve ser considerado como primeiro plano (objeto) até o momento em que haja um gaussiano com evidências suficientes e consistentes que o definam como um novo objeto pertencente ao plano de fundo (STAUFFER et al, 1999).

A figura 34 ilustra o conceito do funcionamento da subtração de fundo por mistura de gaussianas. Neste método, o modelo de plano de fundo é constantemente atualizado utilizando o conceito probabilístico proposto.

Figura 34 - Subtração de fundo por mistura de gaussianas



Fonte: Tabkhi et al (2013)

As distribuições Gaussianas são ordenadas por valores de peso, em que as mais prováveis de ocorrência são priorizadas e recebem maior valor, enquanto que as menos prováveis são colocadas ao final da fila e assumem valores baixos de peso. Sempre que há um novo *pixel*, o mesmo será comparado com cada uma dessas distribuições, iniciando-se pela distribuição mais provável. Um novo *pixel* de valor  $X_t$  é considerado como pertencente à uma distribuição gaussiana específica somente se o seu valor de intensidade estiver dentro de um limite de até 2,5 vezes o desvio padrão da respectiva distribuição. Se o novo valor de *pixel* não

se encaixar em nenhuma das distribuições, a Gaussiana menos provável será substituída por uma nova com valor médio igual ao valor do novo *pixel*, assumindo inicialmente uma elevada variância e com valor baixo de peso anterior associado.

O comportamento das distribuições Gaussianas tende a apresentar as características de regiões da imagem que se referem ao plano de fundo e ao primeiro plano, após o processo de aprendizagem. Objetos da imagem com comportamento estático são representados por *pixels* que se mantêm com um valor praticamente constante e em um longo período de tempo, gerando distribuições compactas e com baixa variância, sendo este o perfil esperado para defini-lo como plano de fundo. Já os objetos em movimento acarretam aos elementos de imagem variações de intensidade que irão gerar inicialmente uma distribuição com alta variância.

O histórico de um pixel em um instante  $t$   $\{X_1, X_2, \dots, X_{t-1}\}$  pode ser modelado pela combinação ponderada de um número  $K$  de distribuições Gaussianas (em geral, define-se  $K$  entre 3 e 5), onde quanto maior a quantidade de distribuições, melhor é a aproximação obtida para a modelagem do pixel. Assim, a probabilidade de observar um pixel com valor de intensidade  $X_t$  é definida pela equação 16:

$$P(X_t) = \sum_{i=0}^K \omega_{i,t} \eta_i(x_{i,t}, \mu_{i,t}, \Sigma_{i,t}) \quad (16)$$

Para cada Gaussiana  $i$  em um instante  $t$ , considera-se  $\omega_{i,t}$  como o peso estimado,  $\mu_{i,t}$  como o valor de média da respectiva distribuição,  $\Sigma_{i,t}$  a matriz de covariância e  $\eta$  é a função densidade de probabilidade normal.

Calcula-se a função densidade de probabilidade normal ( $\eta$ ) a partir da equação 17:

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t - \mu)^T \Sigma^{-1} (X_t - \mu)} \quad (17)$$

Assumindo que as componentes de cor vermelha, verde e azul para cada *pixel* são independentes e de mesma variância, é possível simplificar a inversão da matriz de covariância, o que também resulta na diminuição de custos computacionais. Com isso a equação da matriz de covariância ( $\Sigma_{i,t}$ ) é definida como:

$$\Sigma_{k,t} = \sigma_k^2 \mathbf{I} \quad (18)$$

Sempre que é feita a detecção de um novo *pixel*, realiza-se em seguida a atualização dos parâmetros de peso ( $\omega_{k,t}$ ), média ( $\mu_t$ ) e variância ( $\sigma_t$ ), as quais são utilizadas as seguintes equações:

$$\omega_{k,t} = (1 - \alpha)\omega_{k,t-1} + \alpha(M_{k,t}) \quad (19)$$

Onde  $\alpha$  é a taxa de aprendizagem e  $M_{k,t}$  é 1 para a distribuição em que o valor foi encaixado e 0 para as demais gaussianas.

Os parâmetros de média ( $\mu_t$ ) e variância ( $\sigma_t$ ) se mantêm os mesmos para as distribuições em que o valor do *pixel* não foi correspondente. No entanto, para a distribuição em que o valor do novo *pixel* foi casado, os dois parâmetros se atualizam a partir das equações 20 e 21:

$$\mu_t = (1 - \rho)\mu_{t-1} + \rho X_t \quad (20)$$

$$\sigma_t^2 = (1 - \rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T(X_t - \mu_t) \quad (21)$$

Onde:

$$\rho = \alpha \eta(X_t | \mu_k, \sigma_k) \quad (22)$$

Com a atualização dos parâmetros finalizada, é feita a ordenação das K distribuições Gaussianas do algoritmo, utilizando um fator  $\omega/\sigma$ , para que sejam priorizados os modelos que representam de forma mais efetiva o plano de fundo da imagem. Assim, as primeiras B distribuições são selecionadas como modelo de plano de fundo a partir da equação 23.

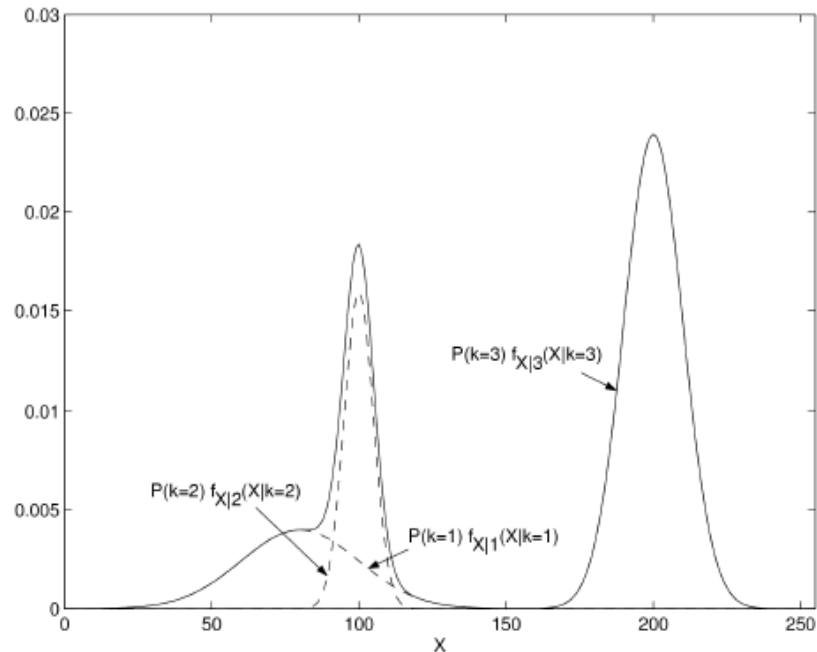
$$B = \underset{b}{\operatorname{argmin}} \left( \sum_{k=1}^b \omega_k > T \right) \quad (23)$$

Onde T é a quantidade mínima de dados (limiar) que deve ser contabilizada para que a distribuição seja considerada plano de fundo.

A figura 35 ilustra as três distribuições gaussianas de um único pixel, onde é possível identificar duas distribuições mais compactadas ( $k=2$  e  $k=3$ ), representando os níveis de intensidade mais frequentes no plano de fundo. A distribuição  $k=1$  possui uma maior variância,

indicando pertencer à um novo nível de intensidade que foi identificado no pixel e com menor probabilidade de ocorrência.

Figura 35 - Distribuições Gaussianas de probabilidade do valor de um pixel



Fonte: Power et al (2002)

Devido à característica de manter o armazenamento de distribuições Gaussianas e a possibilidade de identificar novos modelos que podem se estabelecer conforme o tempo, o método de subtração de fundo por mistura de gaussianas apresenta uma qualidade evidente em relação à adaptabilidade. Novos objetos estáticos que passam a se tornar parte do plano de fundo com o passar do tempo, alterações de iluminação no ambiente e objetos em movimento que fazem parte do plano de fundo (tais como folhas em movimento devido a ação do vento em uma árvore, chuva, neve) são eficientemente identificados pelo algoritmo, possibilitando uma melhor detecção do primeiro plano da imagem. Acharya e Ray (2005) também destacam a vantagem da mistura de gaussianas em armazenar um histórico dos modelos previamente treinados. Quando um *pixel* do primeiro plano passa a fazer parte do plano de fundo, isso não irá ocasionar na eliminação do modelo de plano de fundo existente.

Segundo Zivkovic et al (2006) o processamento computacional da subtração a partir de misturas de gaussianas é impactado pela complexidade da cena, indicando que uma pequena quantidade misturas gaussianas é requisitada para cenários simples (pouca variação no plano de fundo), como por exemplo, em um cenário de tráfego veicular. Logo, é esperado que para

este caso o tempo médio de processamento seja baixo e ainda mais reduzido caso sejam utilizadas poucas distribuições.

Dentre as desvantagens da mistura de gaussianas, está a situação em que um plano de fundo possui variações muito rápidas. Esta característica pode resultar em uma redução na precisão da modelagem, se forem utilizadas poucas distribuições gaussianas, causando problemas para detecção sensível (EL-GAMAL et al, 2012). Adicionalmente, Zivkovic et al (2006) cita a característica de aumento de atribuição de peso conforme a técnica de modelagem recebe sucessivos quadros de objetos em movimento e que se tornaram estáticos na cena no decorrer o tempo, uma característica que pode ser negativa em casos de trânsito congestionado, pois o algoritmo pode reconhecer erroneamente os veículos parados como plano de fundo.

## 1.6 *OPENCV*

O OpenCV (*Open Source Computer Vision Library*) é uma biblioteca de *software* voltada para aplicações de visão computacional e *machine learning*. O OpenCV foi desenvolvido para prover uma infraestrutura comum para aplicações de visão computacional, como também acelerar o uso deste conceito em produtos comerciais. Por se tratar de uma biblioteca *Open Source*, seu código fonte é aberto e de fácil acesso para modificações, possibilitando sua adequação para diferentes negócios.

A biblioteca OpenCV possui mais de 2500 algoritmos otimizados, que podem ser utilizados para aplicações de detecção de imagens em tempo real, por exemplo: na detecção e reconhecimento facial, identificação de objetos, classificação de objetos, rastreamento de movimento em vídeo, detectar imagens correspondentes ou familiares a partir de um banco de imagens pré-estabelecido, dentre outras aplicações. Empresas renomadas como a Google, Microsoft e Honda, são exemplos de companhias que utilizam a biblioteca para aplicações próprias.

Linguagens de programação como Python, C++, Java e MATLAB podem ser utilizadas como interface para essa biblioteca, que também possui suporte para os sistemas operacionais Windows, Android, Mac OS e sistemas baseados em Linux.

## 2 METODOLOGIA

Este trabalho se trata de uma pesquisa aplicada com o objetivo de realizar um estudo acerca de diferentes técnicas de modelagem de plano de fundo para implementação do método de subtração de fundo (*Background Subtraction*) em um ambiente simulado de tráfego veicular, apresentando suas respectivas descrições gerais e características a partir de uma pesquisa bibliográfica. Durante o trabalho também foi realizado o procedimento de pesquisa experimental, ao qual realizou-se o desenvolvimento e aplicação prática dos algoritmos selecionados para objeto de estudo utilizando um banco de imagens para a simulação de um ambiente com veículos em movimentação. O método de abordagem utilizado é a hipotético-dedutiva, onde os dados coletados a partir do experimento prático foram utilizados para análise comparativa e interpretação dos resultados de maneira quantitativa, visando identificar o algoritmo cujo apresentou melhor desempenho quando aplicado em um cenário de trânsito urbano.

O projeto iniciou-se a partir da pesquisa bibliográfica acerca da tendência de novas tecnologias em aplicações de trânsito, dos fundamentos de processamento digital de imagens e das técnicas de modelagem de subtração fundo selecionadas para objeto de estudo, as quais foram: MOG, MOG2, KNN, GMG e CNT. Dentre a bibliografia utilizada para revisão, foram considerados livros relacionados à processamento digital de imagens e visão computacional. Além disso, artigos científicos foram utilizados como referência, com temas relacionados às técnicas de modelagem de plano de fundo selecionadas para estudo e sistemas de transporte inteligente. O conceito geral de cada uma das técnicas modelagem foi descrito no decorrer do trabalho com base nas referências bibliográficas consultadas, como também as características de desempenho previamente descritas para cada um. O conteúdo obtido a partir da pesquisa bibliográfica é utilizado posteriormente como base para análise dos resultados obtidos no experimento prático.

O estágio seguinte da pesquisa consiste em um experimento prático, onde as técnicas de modelagem selecionadas como objeto de estudo foram implementadas utilizando a linguagem de programação Python, em conjunto com o uso da biblioteca OpenCV e um banco de imagens para a simulação de um ambiente de tráfego. O programa desenvolvido efetua a aplicação de cada modelagem de subtração de fundo para o mesmo vídeo e em um mesmo intervalo de tempo. Cada *frame* processado é então gravado com uma identificação única e armazenado em uma pasta para os testes posteriores de acurácia.

Finalizada a etapa de aplicação da subtração de fundo para cada técnica, inicia-se a etapa de coleta de dados de acurácia. Neste momento é feito o desenvolvimento de um segundo algoritmo que realiza o teste de comparação entre as imagens de referência (*Ground Truth*) e as imagens obtidas após a aplicação da subtração de fundo, realizando a comparação *pixel* por *pixel* entre ambas as imagens. Neste caso, o algoritmo faz a coleta de 30 *frames* do grupo de imagens capturados no primeiro algoritmo, e que foram previamente especificados com suas identificações únicas em um vetor denominado *ID*. No programa há um laço de repetição que recebe como parâmetro o tamanho do vetor, onde para cada repetição, é selecionada uma imagem de *Ground Truth* e a imagem que foi obtida a partir do processamento por subtração de fundo, ambas com mesma identificação. As duas são atribuídas a variáveis diferentes (matrizes) e cada elemento referente à um *pixel* é comparado entre as duas imagens, onde dependendo do valor constatado, será selecionado um contador a ser incrementado, que se referem às métricas de acurácia que serão posteriormente calculadas.

Por fim, o teste de desempenho relacionado a tempo de processamento é executado, utilizando um terceiro programa que utiliza a mesma base de código do primeiro algoritmo descrito neste capítulo, possuindo divergências pontuais para realizar a contabilização do tempo decorrido de execução para cada técnica de modelagem.

Os dados obtidos foram coletados e utilizados para calcular as métricas de desempenho para cada método. Estes valores foram posteriormente utilizados para realizar a avaliação dos resultados da pesquisa a partir de uma análise quantitativa.

O experimento prático foi dividido nas seguintes etapas:

- Preparação do ambiente de desenvolvimento;
- Seleção do vídeo;
- Desenvolvimento dos algoritmos de subtração de fundo;
- Coleta de dados de acerto e processamento requisitado.

## 2.1 PREPARAÇÃO DO AMBIENTE DE DESENVOLVIMENTO

Os algoritmos utilizados para implementação das técnicas de modelagem de plano de fundo foram escritos na linguagem de programação *Python*. O ambiente de desenvolvimento utilizado foi a IDE *VSCode*, onde foi realizada a escrita do código e a execução.



Para realizar o desenvolvimento de aplicações relacionadas a visão computacional, recomenda-se a utilização de bibliotecas voltadas para esse tipo de atividade. No caso deste projeto, utilizou-se a biblioteca de código-fonte aberto OpenCV que permitirá a utilização de técnicas de processamento digital de imagens e visão computacional.

A implementação dos algoritmos foi executada por meio de um computador convencional com o programa *VMware Workstation 17 player* para gerar um ambiente de virtualização (máquina virtual). As especificações de *hardware* do computador utilizado para execução dos testes foram as seguintes:

- Sistema operacional: Ubuntu (Arquitetura 64 bits)
- Processador: Intel Core i5-4460 3.20 GHz
- Memória RAM (alocado para máquina virtual): 2GB

## 2.2 DESENVOLVIMENTO DO ALGORITMO DE SUBTRAÇÃO DE FUNDO

O algoritmo realizado neste projeto foi desenvolvido combinando as técnicas fundamentais de processamento digital de imagens que são necessárias para a preparação de cada um dos quadros do vídeo selecionado, seguido da implementação das técnicas de modelagem de plano de fundo, aplicados nos primeiros 5400 quadros do vídeo selecionado. Deste total, o programa realiza a busca dos quadros capturados após processamento e que possuem a numeração especificada de acordo com o vetor *ID*. As diferentes técnicas de modelagem utilizam a mesma lógica de programação e etapas fundamentais de processamento, o que permitiu que um único *script* fosse utilizado para executá-las. O algoritmo desenvolvido, exposto no apêndice A, realiza o processamento de cada um dos *frames* do vídeo selecionado para os testes, seguindo as etapas abaixo:

- Captura de *frames* de vídeo;
- Redimensionamento do quadro;
- Conversão do *frame* de RGB para a escala de cinzas;
- Filtragem espacial para remoção de ruídos;
- Aplicação de subtração de fundo a partir da técnica de modelagem de *background* selecionada;
- Utilização de limiar para binarização da imagem;

- Aplicação de operações morfológicas;
- Apresentação e captura dos *frames* pós-processamento.

O desenvolvimento do código inicia-se com a declaração de variáveis e parâmetros constantes que serão utilizadas no decorrer da lógica de programação. Dentre estes parâmetros, destacam-se *VIDEO\_SOURCE* e *BGS\_TYPE*, onde o primeiro aponta ao endereço do vídeo a ser utilizado para leitura do algoritmo e a segunda variável é utilizada para selecionar a técnica de modelagem que será executada. Em seguida, são declaradas funções que serão utilizadas para o processamento, as quais destacam-se:

- ***getBGSubtractor(BGS\_TYPE)***: Esta função define qual técnica de modelagem será aplicada no *frame*. Nela estão descritas as técnicas que podem ser utilizadas no algoritmo: GMG, MOG, MOG2, KNN, CNT. A modelagem é selecionada de acordo com o valor inteiro especificado para o parâmetro *BGS\_TYPE*.
- ***getFilter(img, filter)***: É a função utilizada para aplicar as operações morfológicas do algoritmo, as quais são: abertura (*'opening'*), fechamento (*'closing'*), dilatação (*'dilation'*), combinação das operações de fechamento, abertura e dilatação (*'combine'*).

Para captura dos *frames*, o programa utiliza como fonte o arquivo de vídeo selecionado para o projeto. O algoritmo entra em um *loop*, onde a cada ciclo é verificado se há a presença ou não de um *frame* para leitura, caso não haja, o *loop* é encerrado e a execução do algoritmo chega ao fim. Para cada ciclo é feita a captura e o processamento de um *frame* do vídeo.

O processamento de imagem do quadro de vídeo capturado é feito por meio da função *cv2.resize()* para redimensionamento da imagem, pelo método *cv2.cvtColor()* para a conversão do *frame* à escala de cinzas e da função *cv2.GaussianBlur()* para filtragem espacial.

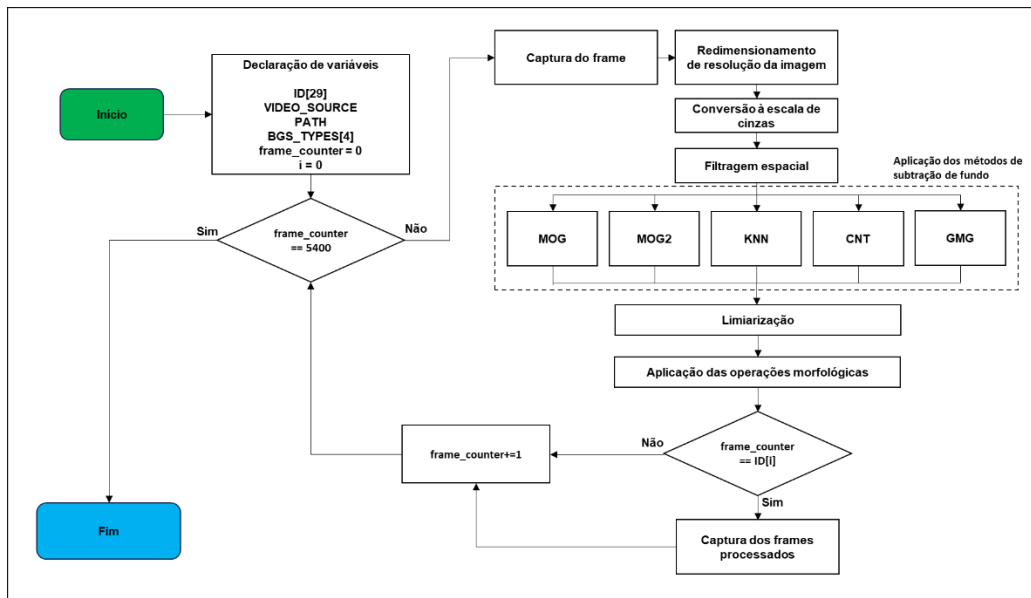
Realizado o processo de filtragem espacial, o programa segue para a implementação do método de subtração de fundo selecionado. As técnicas de modelagem de fundo são atribuídas à diferentes variáveis denominadas *bg\_subtractor*, que é aplicado sob o *frame* cujo passou por filtragem espacial. O resultado de cada subtração de fundo é atribuído às variáveis *bg\_mask*. Após isso, é aplicado um limiar na imagem obtida a partir da função *cv2.threshold()*, resultando em uma imagem binarizada.

O processamento de imagem do *frame* é finalizado ao realizar as operações morfológicas na imagem limiarizada. O programa contém três operações morfológicas:

dilatação (*dilation*), fechamento (*closing*), abertura (*opening*) e uma combinação de todas as operações morfológicas anteriores (*combine*). A opção de operação morfológica a ser aplicada pode ser escolhida pelo usuário. As técnicas de abertura e fechamento são aplicadas pela função *cv2.morphologyEx()* e a de dilatação por meio da função *cv2.dilate()*.

A etapa final do programa é alcançada ao capturar todos os *frames* especificados no vetor *ID*, com a operação de subtração de fundo realizada por cada. Cada quadro processado é capturado e enviado à um diretório específico. O *loop* é encerrado quando todos os primeiros 5400 quadros do vídeo de entrada passam pelo processamento e captura. A figura 36 apresenta um diagrama que representa o funcionamento geral do algoritmo que implementa a subtração de fundo.

Figura 36 - Diagrama representativo do algoritmo utilizado para implementação da subtração de fundo



Fonte: Autoria própria

### 2.3 COLETA DE DADOS PARA MÉTRICAS DE DESEMPENHO

As técnicas de modelagem de plano de fundo selecionadas nesta pesquisa foram testadas para medição de suas métricas após a finalização do estágio de desenvolvimento do código de programação. Os testes foram feitos considerando o mesmo vídeo de entrada que foi selecionado para o projeto, utilizando dois algoritmos para coletar os dados resultantes dos testes. Foram utilizadas duas métricas de desempenho: uma em relação à acurácia da detecção

e outra relacionada ao tempo de processamento necessário. Para os dois casos, foram desenvolvidos programas distintos para realizar a coleta de dados e registro de resultados.

A acurácia foi medida a partir do uso da tabela de contingência (FAWCETT, 2005), cujo apresenta os tipos de resultados que podem ser obtidos em um teste, classificando-os a partir da comparação entre o resultado prático e o ideal, sendo esta uma métrica típica na literatura (CHEUNG, 2003; FERREIRA, 2018) para análise de resultados de detectores de imagem.

Um resultado ideal dentro do escopo deste projeto é a imagem de segmentação ideal, ou seja, uma imagem que representa o resultado esperado com os objetos de interesse destacados, sem presença de ruídos, lacunas em objetos, ou qualquer outro tipo de falha que possa ser presenciada em uma medição. Este tipo de imagem é denominado como *Ground Truth*. A figura 37 apresenta a comparação entre uma imagem de entrada e a sua respectiva imagem com subtração ideal (*Ground Truth*).

Figura 37 - Representação de uma imagem de referência (Ground Truth)



a) Imagem de entrada



b) Imagem com segmentação ideal  
(*Ground Truth*)

Fonte: Autoria própria

A análise foi realizada considerando os *pixels* de primeiro plano (objetos) e plano de fundo do *Ground Truth* como valores positivos (P) e negativos (N), respectivamente.

A classificação dos resultados é realizada conforme abaixo:

- **Verdadeiro Positivo (VP):** *pixels* de primeiro plano (objetos) do *Ground Truth* corretamente detectados como primeiro plano no resultado prático.
- **Falso Positivo (FP):** *pixels* de plano de fundo do *Ground Truth* erroneamente detectados como primeiro plano (objeto) no resultado prático.

- **Verdadeiro Negativo (VN):** *pixels* de plano de fundo do *Ground Truth* corretamente detectados como plano de fundo no resultado prático.
- **Falso Negativo (FN):** *pixels* de primeiro plano (objetos) do *Ground Truth* erroneamente detectados como plano de fundo no resultado prático.

A figura 38 apresenta o conceito de tabela de contingência aplicado ao teste de acurácia realizado neste projeto, onde os positivos equivalem ao valor máximo de nível de intensidade (255, branco) e que representa o primeiro plano (objeto). Os negativos são representados pelo valor mínimo de nível de intensidade (0, preto), representando o plano de fundo.

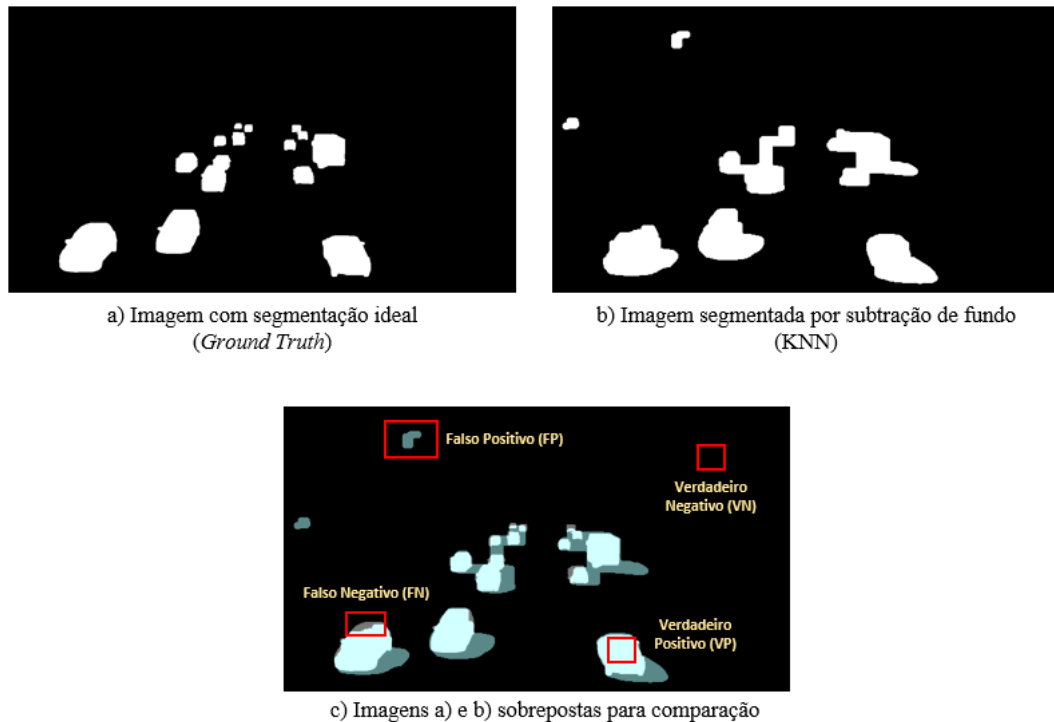
Figura 38 - Tabela de contingência utilizada para o teste de acurácia

	Valor do pixel de referência (Ground Truth)	Valor do pixel obtido (Resultado do teste)
VN	0	0
FP	0	255
FN	255	0
VP	255	255

Fonte: Autoria própria

A figura 39 ilustra a comparação entre uma imagem de *Ground Truth* e o resultado da subtração de fundo por KNN em um *frame* específico, sobrepondo-as e indicando exemplos de elementos de verdadeiros positivos e negativos, como também os falsos positivos e negativos.

Figura 39 - Comparação entre Ground Truth e imagem segmentada por KNN



Fonte: Autoria própria

Os indicadores de acurácia utilizados foram os seguintes:

- **Taxa de verdadeiros positivos (TVP):** indica a quantidade de *pixels* que foram corretamente detectados como primeiro plano, ou seja, é a taxa de acertos na detecção dos *pixels* de primeiro plano.
- **Taxa de falsos positivos (TFP):** indica a quantidade de *pixels* que foram erroneamente detectados como primeiro plano, ou seja, é a taxa de erros na detecção de *pixels* de plano de fundo
- **Taxa de negativos (N):** indica a quantidade de *pixels* detectados do plano de fundo (negativos) em relação à quantidade total de negativos do *Ground Truth*.
- **Taxa de precisão:** indica a proporção de *pixels* de primeiro plano corretamente detectados, em relação à quantidade total de positivos identificados.
- **Taxa de exatidão:** indica a quantidade de *pixels* detectados que estão de acordo com sua respectiva classificação.

Para o cálculo dos indicadores, utilizaram-se as seguintes equações:

$$TVP(\%) = \frac{VP}{P} \cdot 100 \quad (24)$$

$$TFP(\%) = \frac{FP}{N} \cdot 100 \quad (25)$$

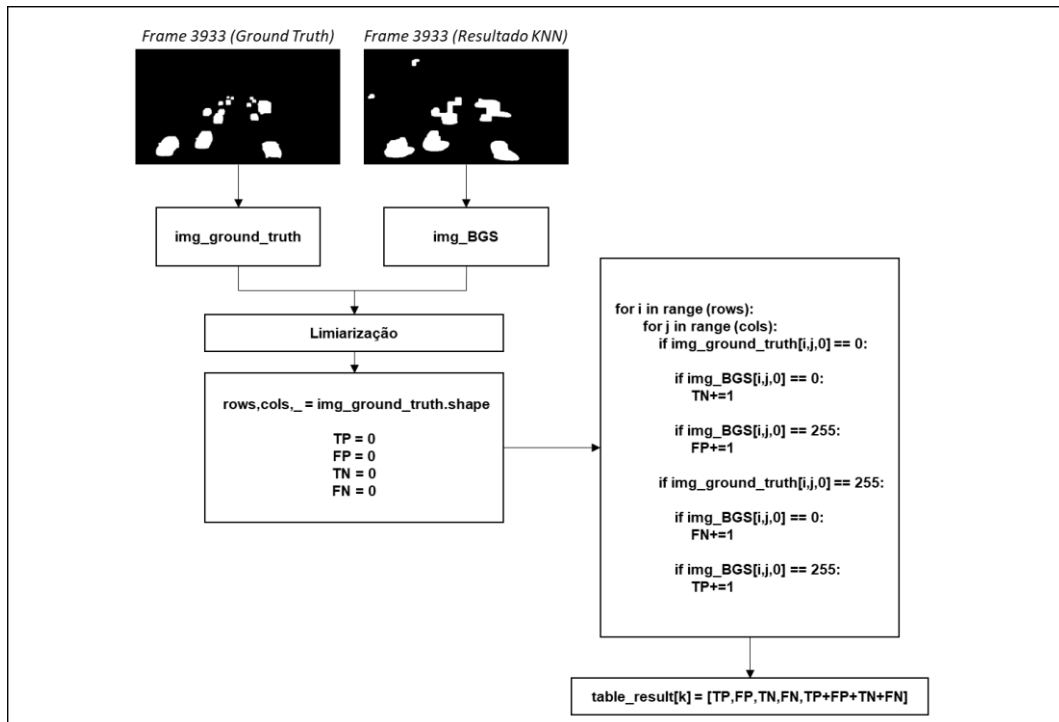
$$N(\%) = \frac{VN}{N} \cdot 100 \quad (26)$$

$$Precisão (\%) = \frac{VP}{VP + FP} \cdot 100 \quad (27)$$

$$Exatidão (\%) = \frac{VP + VN}{P + N} \cdot 100 \quad (28)$$

O programa que realiza o teste de acurácia está exposto no apêndice B, onde este compara uma seleção de 30 imagens de *Ground Truth* com os quadros resultantes da subtração de fundo amostrados do total de 5400 *frames* capturados pelo primeiro algoritmo e selecionados de acordo com as numerações especificadas no vetor *ID*. O algoritmo executa um laço de repetição, recebendo como parâmetro o tamanho do vetor *ID*, onde para cada repetição as variáveis *img\_ground\_truth* e *img\_BGS* recebem uma imagem de *Ground Truth* e uma imagem resultante da subtração de fundo, respectivamente. As variáveis funcionam como matrizes, onde cada posição é referente ao valor de intensidade dos *pixels* das imagens que foram lidas. Em seguida, são feitos dois laços de repetição que atuam realizando a varredura de cada posição dessas matrizes, verificando uma a uma, entre imagem resultante e imagem referência. Com isso, é feita uma comparação entre estes elementos a partir de condicionais que replicam o conceito da tabela de contingência. Cada condição possui um contador que é referente às classificações de resultados determinadas para a análise. Ao final, os dados coletados são enviados para uma tabela (*table\_result*) que é convertida em uma planilha *Excel*. O mesmo processo é realizado para cada uma das técnicas de modelagem abordadas. A figura 40 ilustra o diagrama que representa o funcionamento do algoritmo que faz o teste de *Ground Truth*, aplicado à um *frame* específico que é representado pelo valor do vetor *ID* na posição  $k=23$ .

Figura 40 - Diagrama do algoritmo utilizado para teste de acurácia ( $k = 23$ ,  $ID[k] = 3933$ )



Fonte: Autoria própria

A segunda métrica analisada consiste no tempo de processamento, onde foi calculado o tempo de execução de cada uma das técnicas de modelagem a partir do *script* de teste desenvolvido na linguagem *python* utilizando a biblioteca OpenCV.

O programa utilizado para o teste de desempenho, apresentado no apêndice C, utiliza as mesmas funções básicas utilizadas pelo primeiro algoritmo que realiza subtração de fundo das imagens, seguindo o mesmo formato e divergindo apenas na inclusão de laços de repetição e das funções que coletam os dados de processamento. No *script* os métodos de subtração de fundo são selecionados com base no vetor *BGS\_TYPES* e para cada caso é utilizado um *loop* que fará a cronometragem do tempo de execução do método para um total de 1500 quadros. O ciclo é repetido, totalizando 25 testes para cada técnica de modelagem. A quantidade de testes é equivalente à quantidade de repetições especificada nesse *loop*. Ao finalizá-los para a primeira técnica de modelagem (primeira posição do vetor *BGS\_TYPES*), o contador *i* recebe um acréscimo e os testes iniciam-se novamente, porém para a técnica de modelagem especificada na posição seguinte deste vetor.

O algoritmo utiliza duas funções: *cv2.getTickCount()*, que faz a contagem dos ciclos de *clock* que foram feitos no início e ao final da execução do processamento da imagem, e *cv2.getTickFrequency()*, ao qual apresenta a frequência dos ciclos de *clock*. Todos os valores



contabilizados são atribuídos em uma tabela *table\_result*, onde são armazenados os tempos de processamento.

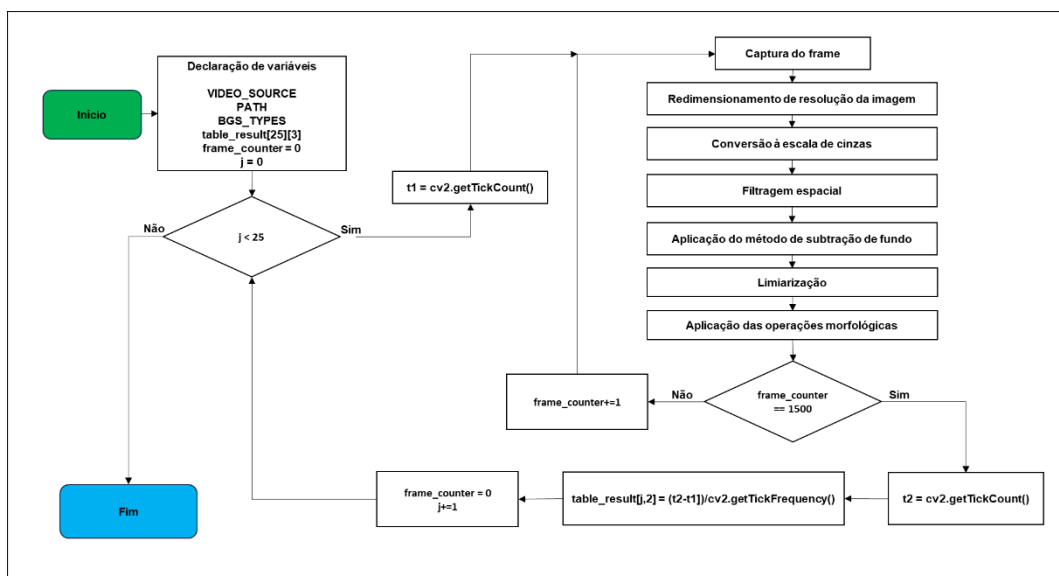
Considerando  $F$  como a frequência obtida a partir da função *cv2.getTickFrequency()*,  $t_1$  e  $t_2$  como a contagem de ciclos de *clock* obtidas pela função *cv2.getTickCount()* no início e no final da execução do processamento da imagem, respectivamente, calculou-se o tempo de processamento requisitado  $t$  a partir da equação 29:

$$t = \frac{t_2 - t_1}{F} \quad (29)$$

O primeiro indicador definido para a métrica de processamento foi o tempo médio necessário para que cada técnica de modelagem realizasse o processamento dos 1500 frames, nas três especificações de resolução estabelecidas. O segundo indicador, tempo médio de processamento por *frame* é o intervalo de tempo médio (em milissegundos) que foi necessário para que cada técnica fosse executada em um único *frame*, também para as três resoluções de vídeo definidas.

A figura 41 demonstra o diagrama de funcionamento do programa para teste de tempo de execução do processamento das imagens. Esta representação demonstra a execução para uma técnica de modelagem de subtração de fundo pré-estabelecida, onde são realizados 25 testes para 1500 *frames*.

Figura 41 - Diagrama do algoritmo utilizado para teste de tempo de processamento



Fonte: Autoria própria

### 3 IMPLEMENTAÇÃO

Neste capítulo é feita a descrição da execução dos processos anteriormente apresentados na metodologia em prática.

#### 3.1 SELEÇÃO DO VÍDEO

Um componente fundamental para a realização do projeto está na utilização de um vídeo que seja adequado ao tipo de cenário a ser simulado. O banco de imagens neste caso se refere à um vídeo (conjunto de quadros) e que deve atender aos requisitos do projeto em questão, apresentando características estáticas, cujo é um ponto fundamental para a aplicação de um método de subtração de fundo. Além disso, é necessário que o vídeo apresente um cenário de tráfego veicular para analisar o desempenho de cada uma das técnicas de modelagem pesquisadas.

Dessa forma, o banco de imagens selecionado para a prática deve atender aos seguintes requisitos:

- Câmera estabelecida em posicionamento fixo;
- Cenário de fundo com característica estática;
- Apresentar veículos em movimento.

Foi utilizado para o projeto prático um exemplar de vídeo que apresenta o tráfego de veículos em uma rodovia de mão dupla, disponibilizado *online* e autorizado pelo autor para uso acadêmico. O vídeo possui duração de 14 minutos.

A figura 42 a seguir apresenta um exemplo dos quadros que compõe o vídeo selecionado para implementação do projeto.

Figura 42 - Amostra de *frame* do vídeo utilizado para o projeto



Fonte: Autoria própria

### 3.2 IMPLEMENTAÇÃO DO ALGORITMO DE SUBTRAÇÃO DE FUNDO

O processamento de imagem dos *frames* do vídeo selecionado inicia-se ao realizar o redimensionamento da imagem para a resolução 640x360, utilizando a função `cv2.resize()`, no que resulta na figura 43. Em seguida é feita a conversão do *frame* à escala de cinzas, utilizando o método `cv2.cvtColor()`. Estas operações permitem um melhor desempenho computacional em termos de processamento. Em seguida, é feita a filtragem espacial na imagem resultante a partir do método `cv2.GaussianBlur()`. A aplicação do filtro espacial tem como objetivo reduzir a quantidade de ruídos que podem estar presentes no *frame* do vídeo em análise.

Figura 43 - Exemplo de *frame* capturado durante o processamento do vídeo



Fonte: Autoria própria

As figuras 44 e 45 apresentam os resultados obtidos após a conversão do *frame* para escala de cinzas e após filtragem espacial, respectivamente.

Figura 44 - Conversão do *frame* capturado de RGB para escala de cinzas



Fonte: Autoria própria

Figura 45 - *Frame* resultante após filtragem espacial



Fonte: Autoria própria

Realizado o processo de filtragem espacial, o algoritmo segue com a aplicação do método de subtração de fundo selecionado. O programa realizou a subtração de fundo separadamente para cada técnica de modelagem e aplicou-se um limiar de  $T = 127$  em cada caso para a binarização da imagem, eliminando a possibilidade do quadro possuir qualquer *pixel* diferente de 0 ou 255, onde o resultado obtido é apresentado na figura 46. Nesse caso:

$$g(x, y) = \begin{cases} 255, & \text{se } f(x, y) \geq 127 \\ 0, & \text{se } f(x, y) < 127 \end{cases}$$

Foi feita uma etapa de calibragem dos parâmetros de subtração de fundo utilizados no programa. Para as técnicas MOG2 e CNT, constatou-se um elevado nível de sensibilidade (alta quantidade de falsos positivos), logo foram feitos ajustes nos parâmetros de *varThreshold* para o valor 300 no caso do MOG2, enquanto que para o CNT houve o ajuste em *maxPixelStability*, alterando-se o intervalo de verificação para 450. Os demais parâmetros das técnicas de modelagem foram estabelecidos de acordo com as especificações da documentação da biblioteca utilizada (OpenCV), seguindo os padrões recomendados pela literatura.

Figura 46 - Imagem binarizada após aplicação de subtração de fundo (MOG)

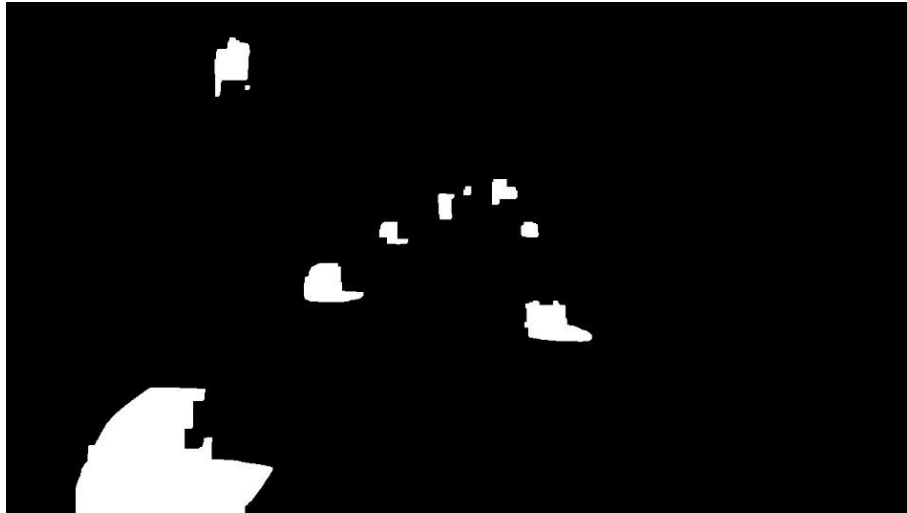


Fonte: Autoria própria

O processamento do *frame* é finalizado ao realizar as operações morfológicas na imagem limiarizada resultante. Nota-se que neste resultado, os objetos de primeiro plano (em maioria, os veículos) apresentam algumas lacunas que compreendem as bordas destes objetos. Com o objetivo de preencher estas lacunas e formar *blobs* (regiões de interesse, formados por um conjunto ou aglomeração de *pixels* do objeto de primeiro plano), é feita a aplicação de operações morfológicas, onde para cada técnica de modelagem foi realizada a mesma operação. A opção selecionada foi a combinação das operações de fechamento, em seguida abertura e por fim dilatação da imagem (no algoritmo, representado pela operação '*combine*') ao qual apresentou um bom desempenho na formação das regiões de interesse do primeiro plano e remoção de ruídos. As operações foram realizadas a partir da chamada do método

`getFilter(bg_mask, 'combine')`, que recebeu como parâmetros a imagem limiarizada e a opção de operação morfológica a ser feita. O resultado obtido após a aplicação das operações morfológicas é apresentado na figura 47.

Figura 47 - Resultado após aplicação de operações morfológicas



Fonte: Autoria própria

A etapa final da implementação do algoritmo é alcançada ao capturar todos *frames* que são referentes imagem subtraída (máscara de primeiro plano), obtida após realização das etapas de processamento de imagem e subtração de fundo. O *loop* é encerrado quando alcançado a meta de processamento de 5400 *frames* do vídeo de entrada. Deste total, são amostrados 30 *frames* de cada técnica de modelagem, em instantes aleatórios, utilizados posteriormente para os testes de acurácia.

A máscara obtida por meio deste algoritmo serve como base para o funcionamento de aplicações de rastreamento de veículos, conforme apresentado na figura 48.

Figura 48 - *Frames* do vídeo de entrada e imagem após processamento por subtração de fundo



a) Imagem de entrada

b) Imagem após aplicação do método de subtração de fundo MOG

c) Representação da detecção de veículos utilizando subtração de fundo

Fonte: Autoria própria

### 3.3 TESTES DE ACURÁCIA

Para a análise de acurácia foi feita a captura de 5400 *frames* em cada técnica de modelagem abordada no estudo, onde foram amostrados 30 quadros aleatórios com instantes de captura equivalentes. A referência utilizada para análise de acurácia dos resultados foram as imagens de *Ground Truth*, construídas manualmente a partir do *software Adobe Photoshop 2023*, totalizando 30 imagens.

Cada um dos quadros processados receberam uma numeração referente ao instante de captura. Essas identificações foram posteriormente utilizadas para auxiliar na seleção das amostras de teste. Ao finalizar a captura dos *frames*, 30 amostras foram coletadas a partir do vetor *ID*, que tem como seus valores os números de identificação das imagens escolhidas. Para este teste, foram selecionados os *frames* com as seguintes identificações:

- $ID = \{87, 169, 193, 307, 391, 475, 517, 613, 643, 871, 1375, 1435, 1597, 1687, 1825, 2016, 2209, 2404, 2467, 2540, 3127, 3284, 3503, 3933, 4210, 4246, 4681, 4830, 5143, 5399\}$

Os quadros selecionados foram comparados com a referência *Ground Truth*, *pixel* por *pixel*, a partir do algoritmo desenvolvido apresentado no Apêndice B e que automatiza essa verificação. Neste teste, o algoritmo captura os valores de intensidade dos elementos de ambas as imagens, comparando-se cada um dos valores de *pixel*, localizados nas mesmas coordenadas entre as duas imagens. Após isso foi aplicada a lógica do conceito da tabela de contingência (FAWCETT, 2005), logo se uma determinada classe é detectada, seu respectivo contador recebe um acréscimo de unidade. Todos os dados coletados foram adicionados em uma matriz (*table\_result*). Ao finalizar o teste, foi gerada uma planilha com os dados de métrica coletados.

O mesmo processo foi realizado para todas as cinco técnicas de modelagem, utilizando 30 amostras de imagem para cada caso, totalizando 150 testes de comparação.

### 3.4 TESTES DE TEMPO DE PROCESSAMENTO

O teste de processamento foi feito utilizando a mesma base do algoritmo de subtração de fundo detalhado na seção 3.2, com a inclusão dos métodos de medição de processamento e um *loop* para realizar a repetição das medições.

Foram realizadas 25 medições para cada método de subtração, onde em cada caso foi feita a cronometragem do tempo necessário para o processamento de um total de 1500 quadros para um mesmo vídeo em três diferentes especificações de resolução: 640x360, 640x480 e 1280x720.

O teste iniciou-se ao realizar a leitura do vídeo e determinando o tipo de resolução a ser testado. Definidos esses parâmetros, o programa pôde ser executado. Neste caso o algoritmo possui dois laços de repetições, o primeiro para selecionar método de subtração a ser medido, enquanto que o segundo estabelece a quantidade de medições a serem realizadas no selecionado. Os dados obtidos foram enviados para uma tabela ao encerrar cada medição. Em seguida, os testes foram feitos novamente para as resoluções restantes e as demais técnicas de modelagem de subtração de fundo.











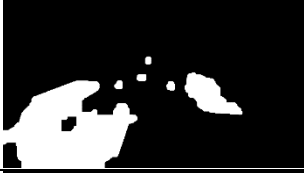
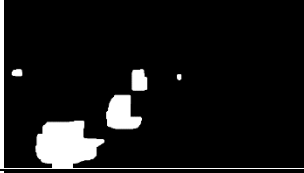
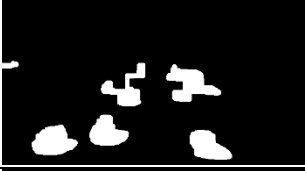
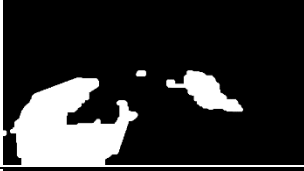







Ao final foram feitos cálculos com base nos dados coletados a partir dos testes realizados, obtendo-se os indicadores de tempo médio de processamento total e por *frame*.



#### 4 RESULTADOS E DISCUSSÕES

Neste tópico serão apresentados os resultados do trabalho que foram obtidos a partir da metodologia utilizada neste projeto. A tabela 1 demonstra exemplos de resultados das imagens segmentadas por subtração de fundo alcançadas no experimento prático, em três diferentes quadros, a partir do algoritmo que foi desenvolvido.

Tabela 1 - Resultados da segmentação das imagens por subtração de fundo

<b>Exemplos de quadros de vídeo coletados para processamento</b>			
	<b>Imagem 1</b>	<b>Imagem 2</b>	<b>Imagem 3</b>
<i>Frame</i>			
<i>Ground Truth</i>			
GMG			
KNN			
MOG			
MOG2			
CNT			

Fonte: Autoria própria

Os elementos de imagem na cor branca representam o primeiro plano, em outras palavras, os objetos em movimento ao qual nesta situação são os veículos. Já os elementos em cor preta, representam os elementos estáticos que compõe o plano de fundo. É possível identificar que para todas as técnicas houve um bom desempenho na detecção do plano de fundo. Em relação aos objetos, nota-se que cada uma das técnicas de modelagem alcançaram um resultado similar ao que de fato é o formato dos veículos em movimento. Em nenhum dos casos foi constatada a detecção e remoção de sombras por parte da técnica de modelagem.

Os algoritmos MOG2 e GMG apresentam baixa presença de falsos positivos no plano de fundo e as detecções dos objetos se assemelham com a imagem de referência (*Ground Truth*), no entanto apresentam lacunas (falsos negativos) ao detectar os objetos. Os algoritmos KNN e MOG apresentaram resultados similares, com um bom desempenho na detecção dos objetos e baixa taxa de falsos positivos no plano de fundo. O algoritmo CNT é o que melhor apresenta detecção dos objetos, aproximadamente sem a presença de falsos negativos, no entanto, demonstra uma grande quantidade de falsos positivos no plano de fundo, ao qual é uma característica indesejável.

#### 4.1 RESULTADOS DO TESTE DE ACURÁCIA

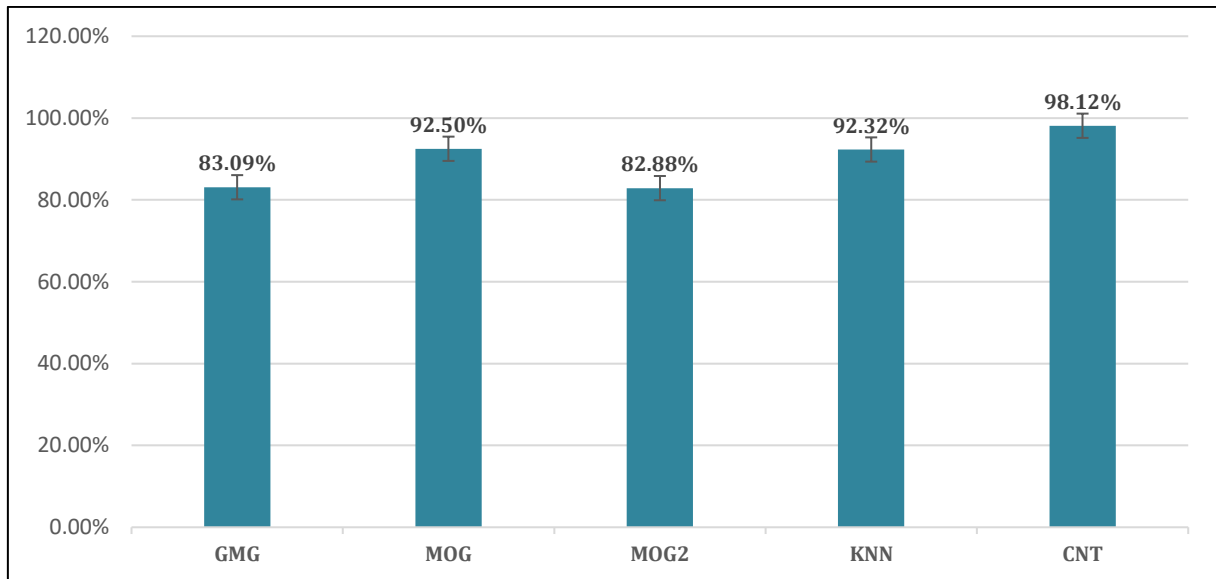
Na tabela 2 estão apresentados os resultados dos testes de acurácia que foram realizados para cada tipo de técnica de modelagem. Para o teste foram selecionados 30 quadros aleatórios de um total de 5400 *frames* processados em cada método. Por fim, calculou-se a média para cada indicador de desempenho das técnicas de modelagem a partir dos dados coletados utilizando as equações 24, 25, 26, 27 e 28.

Tabela 2 - Resultados do teste de acurácia

<b>Técnica de modelagem</b>	<b>TVP (%)</b>	<b>TFP (%)</b>	<b>N (%)</b>	<b>Precisão (%)</b>	<b>Exatidão (%)</b>
GMG	83,09%	1,04%	98,96%	65,66%	98,57%
MOG	92,50%	1,57%	98,43%	61,84%	98,34%
MOG2	82,88%	1,01%	98,99%	66,16%	98,59%
KNN	92,32%	1,47%	98,53%	63,26%	98,41%
CNT	98,12%	3,80%	96,20%	41,24%	96,26%

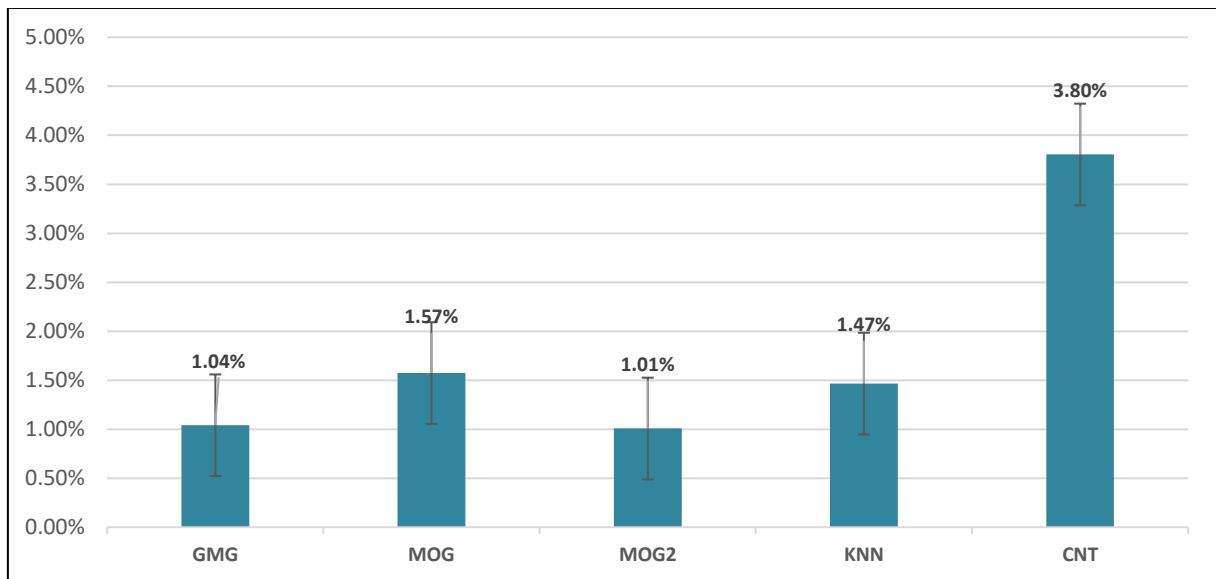
Fonte: Autoria própria

Gráfico 1 - Taxa de Verdadeiros Positivos (TVP) (%)



Fonte: Autoria própria

Gráfico 2 - Taxa de falsos positivos (TFP) (%)



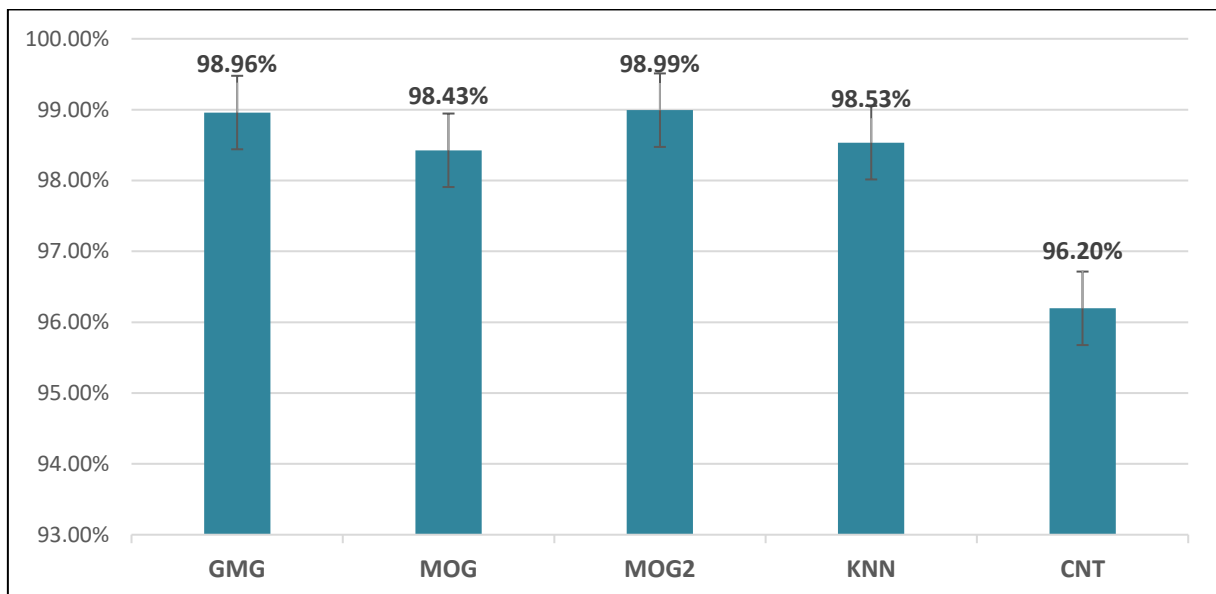
Fonte: Autoria própria

Com base nos resultados obtidos, constata-se que a técnica de modelagem de plano de fundo CNT apresenta a melhor taxa de detecção de primeiro plano, com um valor de TVP de 98,12%. Isso se deve ao fato de que esta técnica apresenta uma elevada sensibilidade, possibilitando detectar facilmente qualquer tipo de movimentação e inclusive pequenos objetos. Apesar disso, a alta sensibilidade do CNT resulta em uma maior presença de ruídos, sendo esta a técnica que apresentou o pior desempenho na detecção de falsos positivos, com uma TFP de 3,80%, um resultado muito elevado e que não é favorável para aplicações de tráfego.

As técnicas de modelagem MOG e KNN apresentam desempenhos similares com TVP igual a 92,50% e 92,32%, respectivamente. Ambas apresentam bons resultados e que são favoráveis para aplicações de tráfego, indicando que possuem a capacidade de identificar objetos móveis em elevada velocidade ao mesmo tempo que não possuem alta suscetibilidade a ruídos do plano de fundo. As situações em que as duas técnicas apresentaram pior desempenho são nas imagens em que os veículos se encontram muito distantes, sendo suprimidos e detectados como plano de fundo (falso negativo) pelos algoritmos. Assim, para uma aplicação de tráfego, recomenda-se que o algoritmo que utilizar estas duas técnicas faça a captura do *frame* a partir de uma região de interesse pré-determinada em que os veículos estejam em uma distância de curta a média distância. Ambas as técnicas apresentaram baixa presença de ruídos, com TFP igual a 1,57% para o MOG e 1,47% para o KNN, indicando um bom desempenho quanto à modelagem de plano de fundo.

Para as técnicas MOG2 e GMG, constataram as menores taxas de TVP e TFP, o que indica uma baixa sensibilidade de ambos os algoritmos. Os piores resultados para esses casos na detecção de *pixels* de primeiro plano foram identificados nas imagens em que a estrada estava sem tráfego, apenas com veículos que se encontravam distantes. Outra característica que justifica o baixo TVP nestes métodos em relação aos demais está falha de modelagem de *background* que pode ser ocasionada quando o algoritmo reconhece incorretamente os veículos em baixa velocidade como elementos de plano de fundo (STAUFFER et al, 1999), ocasionando a presença de lacunas nos objetos detectados (falsos negativos), e que pode ser um resultado desfavorável para aplicações de trânsito em vias que apresentam fluxo congestionado.

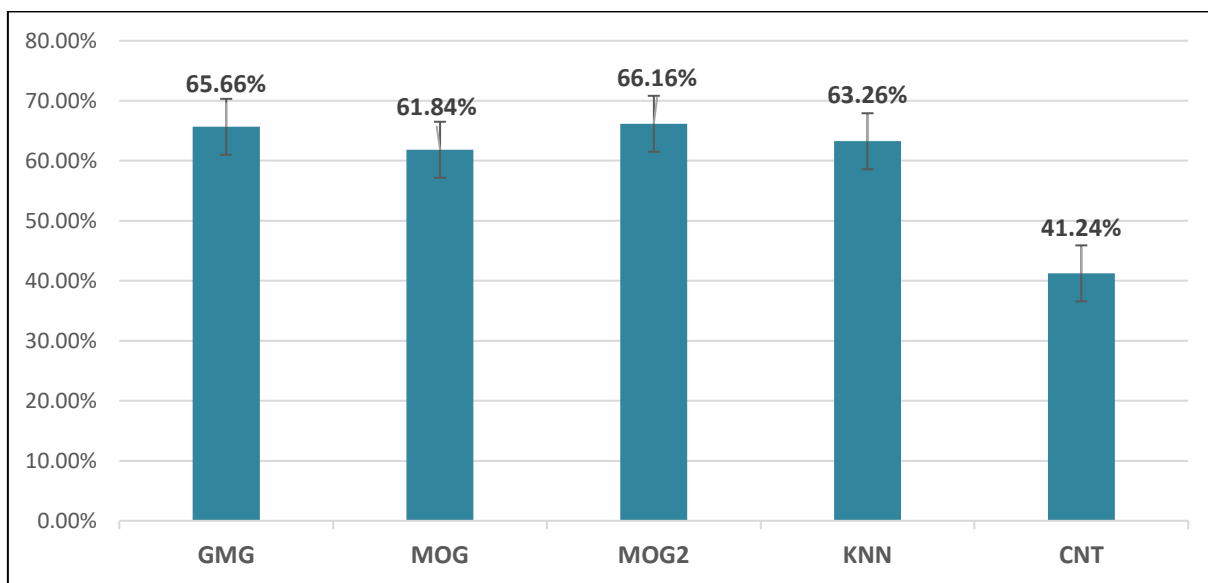
Gráfico 3 - Negativos corretamente detectados - N (%)



Fonte: Autoria própria

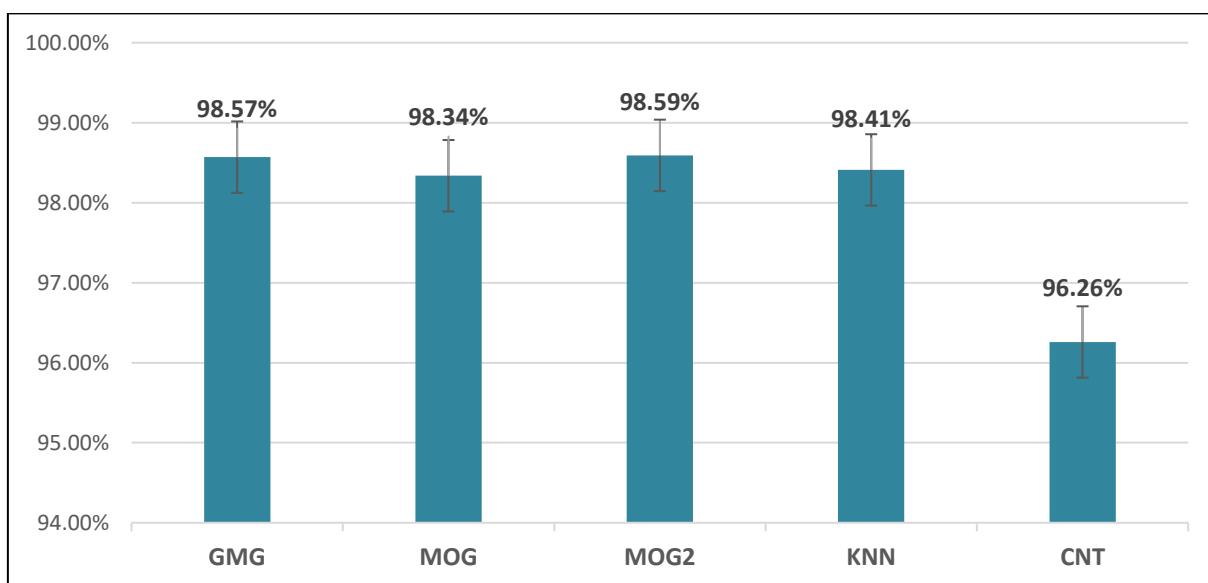
As técnicas de subtração de fundo apresentaram um desempenho elevado em relação à detecção de negativos, indicando que todas os métodos realizaram a modelagem do plano de fundo de forma bem sucedida. Os métodos MOG2 e GMG destacam-se neste quesito, comprovando a característica descrita na literatura em relação à boa adaptabilidade às variações que podem ocorrer no plano de fundo de um cenário (STAUFFER et al, 1999; GODBEHERE, 2012).

Gráfico 4 - Taxa de precisão (%)



Fonte: Autoria própria

Gráfico 5 - Taxa de exatidão (%)



Fonte: Autoria própria

As técnicas MOG2 e GMG apresentaram os melhores resultados em termos de precisão e exatidão. Ambas as técnicas apresentam o melhor desempenho quanto a detecção dos verdadeiros positivos em relação à quantidade total positivos da imagem referência, logo apresentam baixa quantidade de ruídos ao mesmo tempo que realizam um bom desempenho na detecção dos objetos. Além disso estes são os dois algoritmos que apresentam melhor detecção de *pixels* de acordo com sua respectiva classificação. O algoritmo CNT por outro lado foi o caso com pior desempenho devido à sua alta sensibilidade a variações no plano de fundo apresentando alta quantidade de ruídos na sua detecção.

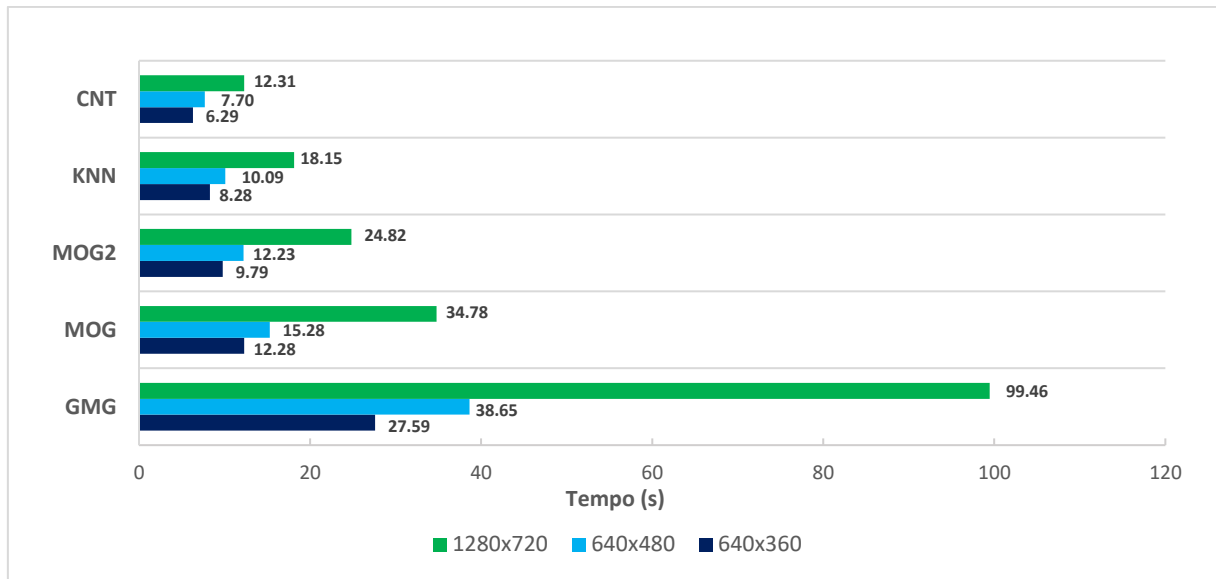
#### 4.2 RESULTADOS DO TEMPO DE PROCESSAMENTO

Nesta seção são apresentados os resultados de tempo médio requisitado por cada técnica de modelagem para o processamento dos *frames* de vídeo. Os dados foram obtidos a partir dos experimentos realizados de acordo com a metodologia descrita, onde foram feitos 25 testes para 1500 *frames* em cada caso, em diferentes resoluções de vídeo, para cada técnica de modelagem descrita. Calculou-se a média para cada critério analisado com base nos dados coletados no experimento. Na tabela 3 estão apresentados os resultados obtidos.

Tabela 3 - Resultados do tempo médio total de processamento para 1500 *frames*

Técnica de Modelagem	Tempo médio de processamento por resolução (em segundos)		
	640x360	640x480	1280x720
GMG	27,59	38,65	99,46
MOG	12,28	15,28	34,78
MOG2	9,79	12,23	24,82
KNN	8,28	10,09	18,15
CNT	6,29	7,70	12,31

Fonte: Autoria própria

Gráfico 6 - Tempo médio requisitado para o processamento de 1500 *frames*

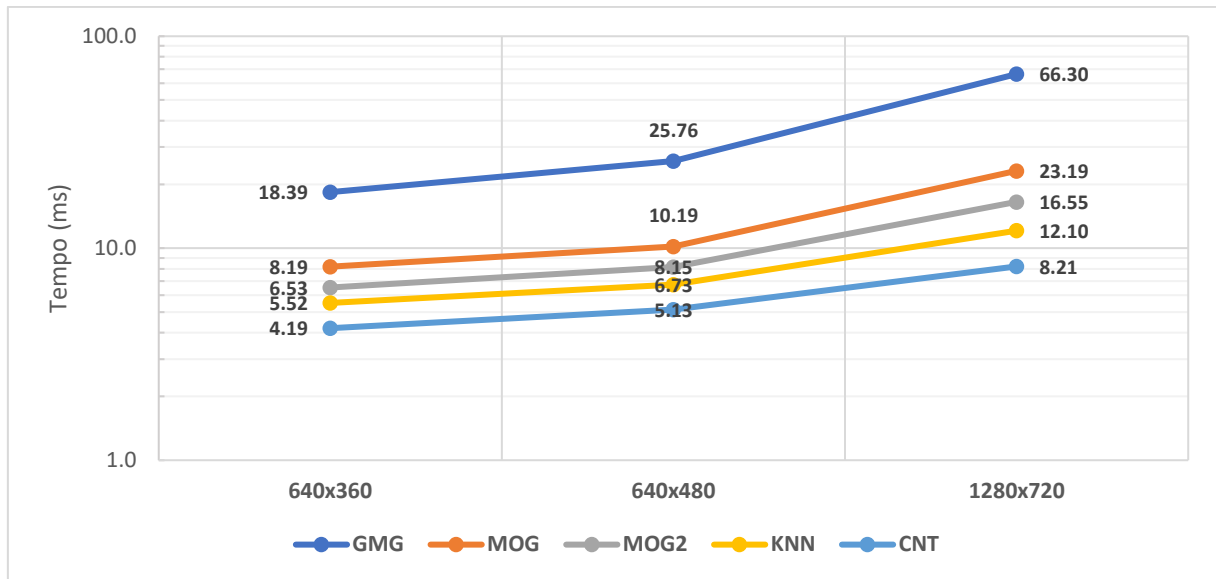
Fonte: Autoria própria

Constata-se com base nos resultados que as técnicas de modelagem de plano de fundo possuem uma tendência de requisitar mais de tempo de processamento conforme há o aumento da resolução do vídeo de entrada, sendo este um fator esperado, tendo em vista que este aumento implica no acréscimo de *pixels* a serem processados. No entanto, cada algoritmo apresentou uma taxa de crescimento diferente. Para o caso da técnica GMG, por exemplo, houve um evidente acréscimo de tempo quando comparados os resultados na resolução de 640x480 para 1280x720, apresentando um elevado aumento de 157,34% no tempo médio requisitado para processamento. Por outro lado, a técnica CNT demonstra um aumento de 59,91%, e quando comparado aos tempos médios de processamento das técnicas anteriormente citadas, esta é a opção que apresentou melhor desempenho neste critério.

Tabela 4 - Resultados do tempo médio de processamento por *frame*

Técnica de Modelagem	Tempo médio de processamento por resolução (em milissegundos)		
	640x360	640x480	1280x720
GMG	18,39	25,76	66,30
MOG	8,19	10,19	23,19
MOG2	6,53	8,15	16,55
KNN	5,52	6,73	12,10
CNT	4,19	5,13	8,21

Fonte: Autoria própria

Gráfico 7 - Tempo médio de processamento por *frame*

Fonte: Autoria própria

As técnicas apresentaram baixos valores de tempo médio para processamento de um *frame*, demonstrando que o método de subtração de fundo é uma alternativa adequada para detecção de objetos em tempo real e também se apresenta como uma possibilidade para sistemas que possuem baixas especificações de *hardware*.

A técnica de modelagem GMG apresentou os maiores tempos médios de processamento dentre os demais casos, isso se justifica pela sua forma de funcionamento, que requer uma quantidade elevada de amostras para que seja feito o treinamento e modelagem do plano de fundo inicial. Além disso, esta é uma técnica recursiva, logo para cada *frame* são aplicadas as técnicas de inferência probabilística e filtragem estabelecidas no algoritmo, resultando em uma maior requisição de tempo para processamento. Como apresentado anteriormente no teste de acurácia, estas etapas utilizadas na modelagem do plano de fundo são eficientes, pois a técnica GMG apresenta um bom resultado em termos de precisão. Por outro lado, esta característica reflete no tempo de processamento, tornando-a uma opção de modelagem mais lenta em comparação aos demais casos estudados, o que a torna desfavorável para aplicações de detecção veicular em tempo real.

As técnicas MOG, MOG2 e KNN apresentaram bons resultados e se demonstram como métodos que realizam o processamento das imagens em tempo hábil, característica que é favorável para a detecção de veículos em tempo real. Os melhores resultados foram obtidos na resolução mais baixa (640x360), onde os algoritmos apresentaram o tempo médio de 8,19 ms (MOG2), 6,53 ms (MOG) e 5,52 ms (KNN) para o processamento de um único *frame*. Em



comparação com as técnicas MOG, MOG2 e GMG, o método KNN é a opção com desempenho mais estável quando aplicado em diferentes resoluções, pois ao comparar seus resultados desde a resolução 640x360 à 640x480, e de 640x480 à 1280x720, este apresenta um aumento no tempo de processamento por *frame* de 21,92% e 79,83%, respectivamente.

A técnica CNT apresenta os melhores resultados em tempo de processamento, sendo esta uma característica esperada de acordo com o que foi proposto para este algoritmo. Além disso, esta é a opção que apresenta a melhor estabilidade em termos de processamento dentre todas as técnicas abordadas, apresentando um acréscimo de processamento por *frame* de 22,36% ao comparar-se os resultados da resolução 640x360 com 640x480 e um acréscimo de 59,91% da resolução 640x480 à 1280x720. O algoritmo CNT apresenta um resultado vantajoso em termos de processamento para aplicações de sistemas de detecção de veículos em tempo real. A velocidade de processamento é devida à abordagem simplificada utilizada pelo algoritmo para a detecção do plano de fundo, onde são aplicadas funções que monitoram o nível de intensidade dos *pixels* e os consideram como plano de fundo ou primeiro plano dependendo do nível de variação destes elementos no decorrer do intervalo de verificação estabelecido. No entanto a sua alta sensibilidade e menor acurácia dentre as demais técnicas a desfavorece neste tipo de aplicação. Além disso, observa-se que métodos com melhor acurácia apresentam velocidade de processamento similares para baixas resoluções, que é o caso quando comparado aos métodos KNN e MOG2, por exemplo. Nestes três casos o tempo médio de processamento por *frame* para resolução 640x360 é aproximado, com os resultados de 4,19 ms para o CNT, 5,52 ms para o KNN e 6,53 ms para o MOG2, sendo estes dois últimos boas opções para detecção e em conjunto com um bom tempo médio de processamento.

## CONCLUSÃO

Neste trabalho foi apresentada a análise de desempenho dos algoritmos de subtração de fundo de uma imagem utilizando métricas de acurácia e tempo de processamento, onde foram selecionados para objeto de estudo as técnicas de modelagem MOG/MOG2, GMG, KNN e CNT aplicados em um ambiente simulado tráfego, visando a identificação do método que se demonstra mais adequado para este cenário, utilizando os resultados medidos no procedimento experimental.

No decorrer do trabalho foi apresentada uma revisão bibliográfica que serviu como base para compreender a aplicabilidade de sistemas de visão computacional em sistemas de trânsito, as operações fundamentais de processamento de imagens digitais e o conceito da operação de subtração de fundo de imagens, onde nesta etapa foram descritos os conceitos gerais de cada uma das técnicas de modelagem recursivas que foram utilizadas no trabalho, possibilitando a compreensão da teoria empregada por estas técnicas para a operação de subtração de plano de fundo. Além disso, estes conceitos possibilitaram a compreensão dos resultados obtidos para cada um dos métodos.

Os resultados práticos obtidos nos experimentos apresentam os comportamentos de cada uma das técnicas de modelagem para subtração de fundo de imagem aplicados em um ambiente de tráfego veicular simulado. Os dados obtidos foram satisfatórios, possibilitando a análise do desempenho de cada um dos métodos e identificar qual a técnica mais adequada para uma aplicação de trânsito. Para esta decisão, realizou-se uma análise quantitativa, onde o algoritmo que apresenta a melhor precisão e exatidão, combinado a um baixo tempo médio de processamento, é considerado a opção mais adequada para uma aplicação de trânsito.

Em relação às demais técnicas analisadas neste estudo, os resultados indicam que os algoritmos CNT e GMG são inadequados para uma aplicação de trânsito. No caso do CNT, constatou-se o pior desempenho em precisão (41,24%) e uma elevada taxa de falsos positivos (3,80%), considerados resultados indesejáveis para uma aplicação de detecção veicular pois demonstra que é uma alternativa sensível à ruídos. Para o GMG, apesar deste ter apresentado uma boa acurácia (precisão de 65,66% e exatidão 98,96%), o seu tempo de processamento foi o indicador decisivo para desconsiderá-lo dentre as demais opções. O algoritmo apresentou um elevado tempo de processamento, demorando em média 25,76 ms para o processamento de um *frame* em um vídeo com resolução 640x480, que é um valor 152,79% maior em comparação ao algoritmo MOG, onde foi constatado o segundo pior desempenho com tempo de processamento por *frame* igual a 10,19 ms e em mesmas condições.

As técnicas de modelagem MOG e KNN apresentaram bons resultados em relação à acurácia e tempo médio de processamento, no entanto, não obtiveram os melhores resultados em taxa de precisão e exatidão dentre as opções analisadas durante o estudo.

A conclusão obtida neste trabalho é a de que o algoritmo MOG2 demonstra-se como a opção mais adequada para uma aplicação de visão computacional voltada para tráfego de veículos, baseada no método de subtração de fundo, pois obteve os melhores resultados nos indicadores selecionados como critérios de decisão, combinado a um tempo de processamento adequado.

Os resultados obtidos para o MOG2 nos indicadores de desempenho decisivos foram os seguintes:

- Taxa de precisão: 66,16%;
- Taxa de exatidão: 98,59%;
- Taxa de falsos positivos: 1,01%;
- Tempo médio de processamento por *frame* (resolução: 640x480): 8,15 ms.

Para trabalhos futuros, é sugerido que a aplicação do método MOG2 seja implementado em um ambiente de tráfego real utilizando um microcomputador de placa única, com baixas especificações de *hardware*, onde as imagens são coletadas a partir de uma câmera de vídeo para captura de imagem em baixa resolução e em tempo real, seguindo as condições de melhor desempenho apresentados pelo algoritmo a partir deste estudo. Com isso, analisar a execução da técnica neste cenário. Outra possibilidade está em realizar pesquisas relacionadas à detecção e extração de sombras em métodos de subtração de fundo, que ocorrem nos veículos em movimento e foram detectadas por todas as técnicas de modelagem, porém, erroneamente consideradas como componentes do primeiro plano (objeto), o que elevou a taxa de falsos positivos para cada um dos métodos estudados.

## REFERÊNCIAS

- ACHARYA, Tinku; RAY, Ajoy. **Image Processing: Principles and Applications**. New Jersey: John Wiley & Sons, Inc., 2005. ISBN 9780471719984.
- BALLARD, Dana; BROWN, Christopher. **Computer Vision**. New Jersey: Prentice-Hall Inc., 1982.
- CHEUNG, S.; KAMATH, C. **A Robust Technique for Background Subtraction in Traffic Video**. *In: SPIE – The International Society for Optical Engineering’s Symposium on Electronic Imaging*, n. 5308, 2004, San Jose, California, United States, pp. 881-892.
- Conselho Nacional de Trânsito – CONTRAN. **Manual Brasileiro de Sinalização de Trânsito: Sinalização Semafórica – Volume 5**. 2014.
- CUCHIARA, Rita; GRANA, Constatino; PICCARDI, Massimo; PRATI, Andrea. **Statistic and knowledge-based moving object detection in traffic scenes**. *In: 2000 IEEE Intelligent Transportation Systems (ITSC2000)*, 2000, Dearborn, Michigan, USA. IEEE, 2000, pp. 27-32.
- CUCHIARA, Rita; GRANA, Constatino; PICCARDI, Massimo; PRATI, Andrea. **Detecting objects, shadows and ghosts in video streams by exploiting color and motion information**. *In: Proceedings 11th International Conference on Image Analysis and Processing*, 2001, Palermo, Italia. IEEE, 2001, pp. 360-365.
- DAMAJ, Issam et al; **Intelligent transportation systems: A survey on modern hardware devices for the era of machine learning**. *In: Journal of King Saud University - Computer and Information Sciences*. ISSN 1319-1578, 2021.
- DAVIS, L.; ROSENFELD A. **Noise Cleaning by Iterated Local Averaging**. *In: IEEE Transactions on Systems, Man, and Cybernetics*, no. 9, 1978. IEEE, vol. 8, pp. 705-710.
- EL-GAMAL, Neven; MOUSTAFA, Hossam; ABOU-CHADI, Fatma. **A new combination method for background subtraction in video sequences**. *In: 8th International Conference on Informatics and Systems (INFOS)*, 2012, Giza, Egypt., pp. MM-21-MM-25.
- FAISCA, B.; DIORIO, R.; **Aplicações da Tecnologia Blockchain em Ambientes de Cidades Inteligentes**. *In: Brazilian Technology Symposium – BTSym Proceedings*. Campinas, São Paulo, Brasil. ISSN 2447-8326. V.1, 2019.
- FAWCETT, T. **An introduction to ROC analysis**. *Pattern Recognition Letters*, vol. 27, pp. 861-874, June 2006.
- FERREIRA, Lucas; CUNTO, Flávio. **Método de subtração de fundo para detecção veicular em condições de tráfego interrompido**. *In: 32º Congresso de Pesquisa e Ensino em Transporte da ANPET*, 2018, Gramado, Rio Grande do Sul, Brasil, pp. 3712-3723.
- FORSYTH, David; PONCE, Jean. **Computer Vision: A Modern Approach**. 2 ed. New Jersey: Pearson, 2012. ISBN 9780136085928.

GODBEHERE, A.; MATSUAKA, A.; GOLDBERG, K.; **Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation.** *In: 2012 American Control Conference (ACC)*, 2012, Montreal, Canada. pp. 4305-4312.

GONZALEZ, Rafael; WOODS, Richard. **Digital Image Processing.** 4 ed. New York: Pearson, 2018. ISBN 9780133356724.

GONZALEZ, Rafael; WOODS, Richard. **Processamento Digital de Imagens.** 3 ed. São Paulo: Pearson, 2010. ISBN 9788576054016.

HARRISON, C.; DONNELLY, I. **A Theory of Smart Cities.** *In: Proceedings of the 55th Annual Meeting of the ISSS – 2011.* Hull, UK, v. 55, n. 1, 23 Sep. 2011.

MARENGONI, Maurício; STRINGHINI, Denise. Tutorial: Introdução à Visão Computacional usando OpenCV. **Revista de Informática Teórica e Aplicada - RITA**, vol. 16, no. 1, pp. 125–160, Mar. 2010.

MARQUES, Ogê; VIEIRA, Hugo. **Processamento Digital de Imagens.** Rio de Janeiro: Brasport, 1999. ISBN 8574520098.

MATCZAK, Grzegorz; MAZUREK, Przemyslaw. Comparative Monte Carlo Analysis of Background Estimation Algorithms for Unmanned Aerial Vehicle Detection. **Remote Sensing**, 2021, vol. 13, p. 870.

POWER, Wayne; SCHOONEES, Johann. **Understanding Background Mixture Models for Foreground Segmentation.** *In: 2002 Image and Vision Computing Proceedings*, 2002, Auckland, New Zealand, pp. 267-271.

QASIM, Salman; KHAN, Kaleem; YU, Miao; KHAN, Muhammad. **Performance Evaluation of Background Subtraction Techniques for Video Frames.** *In: 2021 International Conference on Artificial Intelligence (ICAI)*, 2021, Islamabad, Pakistan, pp. 102-107.

QUEIROZ, J.; GOMES, H. Introdução ao Processamento Digital de Imagens. **RITA**, vol. 13, pp. 11-42, 2006.

SAMADI, Saeed; RAD, Ali; KAZEMI, Farhad; JAFARIAN, Hamed. Performance Evaluation of Intelligent Adaptive Traffic Control Systems: A Case Study. **Journal of Transportation Technologies.** Vol. 2 No. 3, p. 248-259, 2012.

SMITH, Stephen; BARLOW, Gregory; XIE, Xiao-Feng; RUBINSTEIN, Zachary. **Smart Urban Signal Networks: Initial application of the SURTRAC adaptive traffic signal control system.** *In: Proceedings of the 23rd International Conference on Automated Planning and Scheduling*, 2013, Rome, Italy. pp. 434-442.

STAUFFER, C.; GRIMSON, W. **Adaptive background mixture models for real-time tracking.** *In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, proceedings*, 1999, Fort Collins, USA. Vol. 2, pp. 246-252.

TABKHI H.; BUSHEY R.; SCHIRNER G. **Algorithm and architecture co-design of Mixture of Gaussian (MoG) background subtraction for embedded vision.** *In: Asilomar*

**Conference on Signals, Systems and Computers.** Pacific Grove, California, USA, 2013, pp. 1815-1820.

TRNOVSZKÝ, T.; SÝKORA, P.; HUDEC, R. Comparison of background subtraction methods on near Infra-Red spectrum video sequences. **Procedia Engineering**, Amsterdam, Netherlands, vol. 192, pp. 887-892, 2017.

What is computer vision and how does it work with artificial intelligence?. **Algotive**, 2022. Disponível em: <https://www.algotive.ai/blog/what-is-computer-vision-and-how-does-it-work-with-artificial-intelligence>. Acesso em: 17/08/2023

ZIVKOVIC, Z.; Efficient adaptive density estimation per image pixel for the task of background subtraction. **Pattern Recognition Letters**, vol. 27, pp. 773-780, May 2006.

ZIVKOVIC, Zoran; **Improved Adaptive Gaussian Mixture Model for Background Subtraction.** *In: Proceedings of the 17th International Conference on Pattern Recognition*, 2004, Cambridge, UK. Vol. 2, pp. 28-31.

## APÊNDICE A – CÓDIGO PARA IMPLEMENTAÇÃO DE SUBTRAÇÃO DE FUNDO

```

import numpy as np
import cv2
import sys

VIDEO_SOURCE = "/home/leandro/Documents/Visao Computacional/vídeos
/Traffic_3.mp4"
ID =
[87,169,193,307,391,475,517,613,643,871,1375,1435,1597,1687,1825,2016,2209,240
4,2467,2540,3127,3284,3503,3933,4210,4246,4681,4830,5143,5399]
PATH = '/home/leandro/Documents/Visao Computacional/Resultados/Dados
Coletados/BGS/Teste 2/'
frame_counter=0
i = 0

def getKernel(KERNEL_TYPE):
    if KERNEL_TYPE == "dilation":
        kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (3, 3))
    if KERNEL_TYPE == "opening":
        kernel = np.ones((3, 3), np.uint8)
    if KERNEL_TYPE == "closing":
        kernel = np.ones((11, 11), np.uint8)

    return kernel

def getFilter(img, filter):
    if filter == 'closing':
        return cv2.morphologyEx(img, cv2.MORPH_CLOSE, getKernel("closing"),
iterations=2)

    if filter == 'opening':
        return cv2.morphologyEx(img, cv2.MORPH_OPEN, getKernel("opening"),
iterations=2)

    if filter == 'dilation':
        return cv2.dilate(img, getKernel("dilation"), iterations=2)

    if filter == 'combine':
        closing = cv2.morphologyEx(img, cv2.MORPH_CLOSE, getKernel("closing"),
iterations=2)
        opening = cv2.morphologyEx(closing, cv2.MORPH_OPEN,
getKernel("opening"), iterations=2)
        dilation = cv2.dilate(opening, getKernel("dilation"), iterations=2)

```

```

        return dilation

def getBGSubtractor(BGS_TYPE):
    if BGS_TYPE == "GMG":
        return
cv2.bgsegm.createBackgroundSubtractorGMG(initializationFrames=120,
decisionThreshold=0.8)
    if BGS_TYPE == "MOG":
        return cv2.bgsegm.createBackgroundSubtractorMOG(history=200,
nmixtures=5, backgroundRatio=0.7, noiseSigma=0)
    if BGS_TYPE == "MOG2":
        return cv2.createBackgroundSubtractorMOG2(history=500,
detectShadows=True, varThreshold=300)
    if BGS_TYPE == "KNN":
        return cv2.createBackgroundSubtractorKNN(history=500,
detectShadows=True, dist2Threshold=400)
    if BGS_TYPE == "CNT":
        return cv2.bgsegm.createBackgroundSubtractorCNT(minPixelStability=15,
useHistory=True,
                                                    maxPixelStability=15*3
0, isParallel=True)
    print("Detector inválido")
    sys.exit(1)

cap = cv2.VideoCapture(VIDEO_SOURCE)
hasFrame, frame = cap.read()

bg_subtractor_GMG = getBGSubtractor("GMG")
bg_subtractor_MOG = getBGSubtractor("MOG")
bg_subtractor_MOG2 = getBGSubtractor("MOG2")
bg_subtractor_KNN = getBGSubtractor("KNN")
bg_subtractor_CNT = getBGSubtractor("CNT")

while (cap.isOpened()):
    ok, frame = cap.read()
    if not ok:
        print("Erro")
        break

    frame_resize = cv2.resize(frame,(0,0),fx=0.5,fy=0.5)
    frame_gray = cv2.cvtColor(frame_resize,cv2.COLOR_BGR2GRAY)
    frame_gray_filtered = cv2.GaussianBlur(frame_gray,(3,3),5)

    bg_mask_GMG = bg_subtractor_GMG.apply(frame_gray_filtered)
    bg_mask_MOG = bg_subtractor_MOG.apply(frame_gray_filtered)
    bg_mask_MOG2 = bg_subtractor_MOG2.apply(frame_gray_filtered)
    bg_mask_KNN = bg_subtractor_KNN.apply(frame_gray_filtered)
    bg_mask_CNT = bg_subtractor_CNT.apply(frame_gray_filtered)

```



```
(T, bg_mask_GMG) = cv2.threshold(bg_mask_GMG,127,255,cv2.THRESH_BINARY)
(T, bg_mask_MOG) = cv2.threshold(bg_mask_MOG,127,255,cv2.THRESH_BINARY)
(T, bg_mask_MOG2) = cv2.threshold(bg_mask_MOG2,127,255,cv2.THRESH_BINARY)
(T, bg_mask_KNN) = cv2.threshold(bg_mask_KNN,127,255,cv2.THRESH_BINARY)
(T, bg_mask_CNT) = cv2.threshold(bg_mask_CNT,127,255,cv2.THRESH_BINARY)

morph_filter = 'combine'
bg_mask_GMG = getFilter(bg_mask_GMG, morph_filter)
bg_mask_MOG = getFilter(bg_mask_MOG, morph_filter)
bg_mask_MOG2 = getFilter(bg_mask_MOG2, morph_filter)
bg_mask_KNN = getFilter(bg_mask_KNN, morph_filter)
bg_mask_CNT = getFilter(bg_mask_CNT, morph_filter)

if (frame_counter == ID[i]):

    save_frame_GMG = cv2.imwrite(PATH+'GMG
BGS/GMG_'+str(frame_counter)+'.png',bg_mask_GMG)
    save_frame_MOG = cv2.imwrite(PATH+'MOG
BGS/MOG_'+str(frame_counter)+'.png',bg_mask_MOG)
    save_frame_MOG2 = cv2.imwrite(PATH+'MOG2
BGS/MOG2_'+str(frame_counter)+'.png',bg_mask_MOG2)
    save_frame_KNN = cv2.imwrite(PATH+'KNN
BGS/KNN_'+str(frame_counter)+'.png',bg_mask_KNN)
    save_frame_CNT = cv2.imwrite(PATH+'CNT
BGS/CNT_'+str(frame_counter)+'.png',bg_mask_CNT)
    i+=1

    frame_counter+=1
    print(frame_counter)

if (cv2.waitKey(1) & 0xFF == ord("q")) or frame_counter == 5400:
    break

cap.release()
cv2.destroyAllWindows()
```

## APÊNDICE B – CÓDIGO PARA TESTES DE *GROUND TRUTH*

```

import numpy as np
import pandas as pd
import cv2

table_result = np.zeros((30,5), dtype=int)
ID =
[87,169,193,307,391,475,517,613,643,871,1375,1435,1597,1687,1825,2016,2209,240
4,2467,2540,3127,3284,3503,3933,4210,4246,4681,4830,5143,5399]
PATH = '/home/leandro/Documents/Visao Computacional/Resultados/Dados
Coletados/Ground Truth/'

for k in range (len(ID)):
    img_ground_truth = cv2.resize(cv2.imread(PATH+'Reference/GT-
frame_'+str(ID[k])+'.png'),(0,0),fx=0.5,fy=0.5)
    img_BGS = cv2.imread(PATH+'BGS Samples/Teste
2/MOG2/MOG2_'+str(ID[k])+'.png')
    (T, img_ground_truth) =
cv2.threshold(img_ground_truth,127,255,cv2.THRESH_BINARY)
    (T, img_BGS) = cv2.threshold(img_BGS,127,255,cv2.THRESH_BINARY)

    rows,cols,_ = img_ground_truth.shape
    TP = 0
    FP = 0
    TN = 0
    FN = 0

    for i in range (rows):
        for j in range (cols):
            if img_ground_truth[i,j,0] == 0:

                if img_BGS[i,j,0] == 0:
                    TN+=1

                if img_BGS[i,j,0] == 255:
                    FP+=1

            if img_ground_truth[i,j,0] == 255:

                if img_BGS[i,j,0] == 0:
                    FN+=1

                if img_BGS[i,j,0] == 255:
                    TP+=1

```

```
table_result[k] = [TP,FP,TN,FN,TP+FP+TN+FN]

print('\nImagen', k, ': ')
print('TP', TP)
print('FP', FP)
print('TN', TN)
print('FN', FN)
print('Total pixels', TP+FP+TN+FN)

print('\n\n',table_result)
df = pd.DataFrame(table_result).T
df.to_excel(excel_writer = PATH+'/Data/Teste/GroundTruth.xlsx')
```

## APÊNDICE C – CÓDIGO PARA TESTE DE TEMPO DE PROCESSAMENTO

```

import numpy as np
import pandas as pd
import cv2
import sys

VIDEO_SOURCE = "/home/leandro/Documents/Visao
Computacional/videos/Traffic_3.mp4"
PATH = '/home/leandro/Documents/Visao Computacional/Resultados/Dados
Coletados/Teste de Desempenho'
BGS_TYPES = ["GMG", "MOG", "MOG2", "KNN", "CNT"]
np.set_printoptions(precision = 3, suppress = True)
frame_counter = 0
k=0

def getKernel(KERNEL_TYPE):
    if KERNEL_TYPE == "dilation":
        kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (3, 3))
    if KERNEL_TYPE == "opening":
        kernel = np.ones((3, 3), np.uint8)
    if KERNEL_TYPE == "closing":
        kernel = np.ones((11, 11), np.uint8)
    return kernel

def getFilter(img, filter):
    if filter == 'closing':
        return cv2.morphologyEx(img, cv2.MORPH_CLOSE, getKernel("closing"),
iterations=2)

    if filter == 'opening':
        return cv2.morphologyEx(img, cv2.MORPH_OPEN, getKernel("opening"),
iterations=2)

    if filter == 'dilation':
        return cv2.dilate(img, getKernel("dilation"), iterations=2)

    if filter == 'combine':
        closing = cv2.morphologyEx(img, cv2.MORPH_CLOSE, getKernel("closing"),
iterations=2)
        opening = cv2.morphologyEx(closing, cv2.MORPH_OPEN,
getKernel("opening"), iterations=2)
        dilation = cv2.dilate(opening, getKernel("dilation"), iterations=2)
        return dilation

```

```

def getBGSubtractor(BGS_TYPE):
    if BGS_TYPE == "GMG":
        return
cv2.bgsegm.createBackgroundSubtractorGMG(initializationFrames=120,
decisionThreshold=0.8)
    if BGS_TYPE == "MOG":
        return cv2.bgsegm.createBackgroundSubtractorMOG(history=200,
nmixtures=5, backgroundRatio=0.7, noiseSigma=0)
    if BGS_TYPE == "MOG2":
        return cv2.createBackgroundSubtractorMOG2(history=500,
detectShadows=True, varThreshold=300)
    if BGS_TYPE == "KNN":
        return cv2.createBackgroundSubtractorKNN(history=500,
detectShadows=True, dist2Threshold=400)
    if BGS_TYPE == "CNT":
        return cv2.bgsegm.createBackgroundSubtractorCNT(minPixelStability=15,
useHistory=True,
                                                    maxPixelStability=15*3
0, isParallel=True)

    print("Detector inválido")
    sys.exit(1)

for i in range (len(BGS_TYPES)):
    table_result = np.zeros((25,3))
    for j in range (25):
        cap = cv2.VideoCapture(VIDEO_SOURCE)
        hasFrame, frame = cap.read()
        bg_subtractor = getBGSubtractor(BGS_TYPES[i])
        table_result[j,0] = cv2.getTickCount()

        while (cap.isOpened()):
            ok, frame = cap.read()
            if not ok:
                print("Erro")
                break

            frame_resize = cv2.resize(frame,(0,0),fx=0.5,fy=0.5)
            frame_gray = cv2.cvtColor(frame_resize,cv2.COLOR_BGR2GRAY)
            frame_gray_filtered = cv2.GaussianBlur(frame_gray,(3,3),5)
            bg_mask = bg_subtractor.apply(frame_gray_filtered)
            (T, bg_mask) = cv2.threshold(bg_mask,127,255,cv2.THRESH_BINARY)
            bg_mask = getFilter(bg_mask, 'combine')

            if (cv2.waitKey(1) & 0xFF == ord("q")) or frame_counter == 1500:
                table_result[j,1] = cv2.getTickCount()
                break

```

```
        frame_counter+=1

        table_result[j,2] = (table_result[j,1]-
table_result[j,0])/cv2.getTickFrequency()
        frame_counter = 0

        print('Resultado '+BGS_TYPES[i]+':\n', table_result)
        df = pd.DataFrame(table_result).T
        df.to_excel(excel_writer = PATH+'/' +BGS_TYPES[i]+' /Teste-
Desempenho_'+BGS_TYPES[i]+' .xlsx')

cap.release()
cv2.destroyAllWindows()
```