

UNIVERSIDADE NOVE DE JULHO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIDADES INTELIGENTES E
SUSTENTÁVEIS

ROGÉRIO SANTINO BARBOZA

VISÃO COMPUTACIONAL:

ESTUDO COMPARATIVO DE ALGORITMOS DE SUBTRAÇÃO DE FUNDO
APLICADOS EM SOLUÇÕES PARA O GERENCIAMENTO DE TRÁFEGO URBANO
DE VEÍCULOS E PEDESTRES

SÃO PAULO

2023

ROGÉRIO SANTINO BARBOZA

VISÃO COMPUTACIONAL:

ESTUDO COMPARATIVO DE ALGORITMOS DE SUBTRAÇÃO DE FUNDO
APLICADOS EM SOLUÇÕES PARA O GERENCIAMENTO DE TRÁFEGO URBANO
DE VEÍCULOS E PEDESTRES

COMPUTER VISION:

COMPARATIVE STUDY BETWEEN BACKGROUND SUBTRACTION ALGORITHMS
APPLIED IN SOLUTIONS FOR URBAN TRAFFIC MANAGEMENT OF VEHICLES
AND PEDESTRIANS

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Cidades Inteligentes e Sustentáveis da Universidade Nove de Julho – UNINOVE, como requisito parcial para obtenção do grau de **Mestre em Cidades Inteligentes e Sustentáveis**.

Orientador: Prof. Dr. João Alexandre Paschoalin Filho

São Paulo
2023

FICHA CATALOGRÁFICA

Barboza, Rogério Santino.

Visão computacional: estudo comparativo de algoritmos de subtração de fundo aplicados em soluções para o gerenciamento de tráfego urbano de veículos e pedestres. / Rogério Santino Barboza. 2023.

280 f.

Dissertação (Mestrado) - Universidade Nove de Julho - UNINOVE, São Paulo, 2023.

Orientador (a): Prof. Dr. João Alexandre Paschoalin Filho.

1. Visão computacional. 2. Detecção de objetos. 3. OpenCV. 4. Cidades inteligentes. 5. Tráfego urbano.

VISÃO COMPUTACIONAL:
ESTUDO COMPARATIVO DE ALGORITMOS DE SUBTRAÇÃO DE FUNDO
APLICADOS EM SOLUÇÕES PARA O GERENCIAMENTO DE TRÁFEGO URBANO
DE VEÍCULOS E PEDESTRES

POR
ROGÉRIO SANTINO BARBOZA

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Cidades Inteligentes e Sustentáveis da Universidade Nove de Julho – UNINOVE, como requisito parcial para obtenção do grau de Mestre em Cidades Inteligentes e Sustentáveis, apresentada à Banca Examinadora formada por:

Prof. Dr. João Alexandre Paschoalin Filho – PPG-CIS – Universidade Nove de Julho – UNINOVE

Prof. Dr. Cleber Gustavo Dias – PPGI – Universidade Nove de Julho – UNINOVE

Prof. Dr. Cristiano Capellani Quaresma – PPG-CIS – Universidade Nove de Julho – UNINOVE

São Paulo, 24 de fevereiro de 2023

Dedico este trabalho à Anália Santino Barboza, representante da força e determinação feminina, nordestina, mulata, mãe, solteira, guerreira, humilde e batalhadora, que passou fome para ter o que alimentar seus filhos em São Paulo a maior e mais rica cidade do país; ensinou o valor do trabalho e determinação. Espero com este trabalho acadêmico, perpetuar sua memória e lições, que embora não tenha aprendido em sala de aula, aprendi com a melhor professora! Obrigado mãe.

Dividindo essa jornada e ensinando-me a nunca desistir e a perseverar; não importando o quanto seja difícil essa batalha, à Viviane Salles Cabral, que em meio ao “inferno na terra” que viveu durante a pandemia e apesar de suas perdas pessoais; conseguiu encontrar espaço para me ensinar o lado doce da vida! Ensinar que até em um deserto, há vida! Há sempre esperanças de uma vida melhor! Sem seu incansável apoio, não teria ingressado nessa jornada. Obrigado minha “pimentinha”, sem você minha vida ainda seria em preto e branco, agora, tudo está em alta resolução e colorido.

E a por fim, mas não menos importante, a você que lê esse trabalho, peço que se lembre de uma das Odes de Horácio Flaco (65 a.C.- 8 a.C.) e experimente o “*CARPE DIEM QUAM MINIMUM CREDULA PÓSTERO*” ou seja “*aproveite o dia de hoje e confie o mínimo possível no amanhã*” e o meu obrigado por prestigiar-me.

AGRADECIMENTOS

Meus agradecimentos aos profissionais envolvidos no Programa de Pós-Graduação em Cidades Inteligentes e Sustentáveis por diversos momentos de trocas de conhecimento e lições de vida:

- Iniciando pelos professores que participaram do processo seletivo do programa; sem seus olhares, não seria possível ter vivenciado essa jornada única e desafiadora.

- Ao meu orientador, Dr. João Alexandre Paschoalin Filho, pelos ensinamentos, colaboração e pela paciência, a cada reunião, uma nova oportunidade de fazer mais e melhor!

- Às professoras: Dra. Alessandra Cristina Guedes Pellini, Dra. Amarilis Lucia Casteli Figueiredo Gallardo, Dra. Cintia Elisa de Castro Marino, Dra. Heidy Ramos Rodriguez, Dra. Tatiana Tucunduva Philippi Cortese e aos professores: Dr. Antonio Pires Barbosa, Dr. Cristiano Capellani Quaresma, Dr. José Eduardo Storópoli, Dr. Luis Fernando Massonetto e Dr. Wilson Levy Braga da Silva Neto, que transferiram sua paixão e dedicação à ciência e a arte de ensinar.

- Em especial, ao professor Dr. Leonardo Vils, que além da sua insaciável fome pelo conhecimento científico, transferiu seu apresso pela vida e família; à arte de viver e experimentar uma boa refeição e apreciar uma boa bebida, seja sozinho, com os amigos ou em família. Transferiu mais do que precisava e por este motivo agradeço não somente pelas experiências educacionais, mas pelos valores pessoais: do simples bate-papo desprendido e verdadeiro; ao amor pela ciência e experiências; além das receitas gastronômicas. Meu muito obrigado Leo e à sua boina!

- À colaboradora do Programa de Pós-Graduação em Cidades Inteligentes e Sustentáveis: Suely Maria Iafrate, pela paciência e auxílio nos processos acadêmicos, seja de caráter pessoal ou como representante discente da querida turma PPG-CIS 2021;

- À instituição UNINOVE e a sua administração, que em sua estrutura interna promove a todos os alunos, independentemente de qualquer outro processo: a igualdade humana, o acesso a ciência e cultura, a infinita sede pelo conhecimento e o acesso a este universo infinito. Agradeço a minha bolsa integral do curso, que sem esse apoio, não seriam possíveis o meu ingresso e a minha experiência.

- Aos demais professores que porventura possa não ter mencionado e aos do módulo internacional da Universidade de Haifa em 2021 que demonstraram também o seu apresso e dedicação à ciência e a transferência de conhecimento sem fronteiras.

- Aos membros da banca, que em ambas as ocasiões, qualificação e defesa, avaliaram e propuseram as correções e melhorias na forma como esse trabalho se apresenta; contribuindo com meu aprendizado inclusive.

- A todos os diretores e principalmente ao Sr. Ricardo Lawrence Durski, sócio-diretor da Prosys Tecnologia, empresa ao qual estive presente durante o curso e que sem essa contribuição e colaboração, não teria em muitos momentos, condições de dedicar-me a essa jornada acadêmica.

- Ao amigo Willians Pereira Dutra, que me apresentou a solução *Yolo*, além de sempre colocar-se a disposição para ajudar nos esclarecimentos práticos acerca do tema.

- À Sister Rosetta Tharpe (1915-1973) que uniu a música gospel e acelerou o blues um pouco antes dos anos 40, influenciando Little Richards e Elvis Presley e é conhecida como a criadora e a Albert James Freed (1921-1965), que em 1951 em Cleveland, Ohio, tocou *Rhythm & Blues* para uma audiência multirracial e utilizou uma expressão para descrevê-la; sim, é disso mesmo que falo, vida longa e prospera ao “*Rock & Roll*”, que me influencia e acompanha durante toda a vida.

E, finalmente, aos nobres colegas de jornada, aos quais compartilhamos alegrias e tristezas, como uma família, que mesmo distante fisicamente por conta da pandemia de COVID-19; estivemos unidos e colaborando com o desenvolvimento individual e coletivo. Somando desta forma, ciência e dedicação.

"No que diz respeito ao desempenho, ao compromisso, ao esforço, à dedicação, não existe meio termo. Ou você faz uma coisa bem-feita ou não faz."

(Ayrton Senna)

RESUMO

O crescimento da população e da frota de veículos em todo o mundo, principalmente nas grandes cidades, provoca o aumento dos congestionamentos, gerando complicações no trânsito. As principais soluções para reduzir o congestionamento requerem tempo, que é caro e ineficaz a médio e longo prazo. O monitoramento inteligente de tráfego é um ramo dos sistemas de transporte inteligente que se concentra em melhorar as condições do tráfego urbano. O principal objetivo de tais sistemas é melhorar o sistema de tráfego, reduzindo os problemas de fluidez dos veículos e melhorando a mobilidade urbana. Neste contexto, surge a necessidade de melhorar e automatizar o sistema de semáforos existente. Estabelecer uma técnica mista de inteligência artificial e visão computacional pode ser desejável para desenvolver um sistema de tráfego confiável e escalável que possa ajudar a resolver esses problemas. Este trabalho tem como foco a utilização da tecnologia de visão computacional como ferramenta para a construção de um sistema com recursos para análise de tráfego síncrona e automatizada, para determinar qual algoritmo a ser aplicado, é comparado o desempenho de diferentes algoritmos de subtração de fundo (BGS) em um sistema prático que aplica visão computacional para identificar e rastrear veículos e ser capaz de detectar pedestres é apresentado. O software foi gerado para analisar algoritmos BGS, conta ainda, com contagem de veículos, registro de veículos e sentido de direção, divididos em dois grupos: carro e caminhão. Em outro software, um aplicativo específico para pedestres. Com isso, foi possível determinar quais algoritmos podem ser mais efetivamente adequados a um sistema mais complexo, como o gerenciamento de tráfego urbano, por meio da adoção de dispositivos IoT na borda ou em veículos autônomos que utilizam de alguma forma sistemas de visão computacional — uma interface gráfica foi disponibilizada para auxiliar no estudo destes algoritmos. A pesquisa tem como objetivo através da Revisão Sistemática da Literatura (RSL) de artigos publicados no período de 01/01/1970 a 31/01/2023, através de trabalhos relacionados ao monitoramento do tráfego urbano de pedestres e veículos diversos. Como o BGS é uma técnica utilizada em visão computacional para detectar objetos em movimento em uma sequência de imagens obtidas de um vídeo, por exemplo, verificou-se que contar apenas com o BGS sem nenhum outro processamento ou suporte por aprendizado de máquina (ML) ou deep learning (DL), não apresentam acurácia o suficiente para aplicação direta em sistemas de semáforos ou monitoramento autônomo. Porém, com a constante evolução de softwares e hardwares especializados, a aplicação conjunta dessas soluções é promissora. Como demonstram os resultados deste trabalho, existe a necessidade de melhorar os filtros individuais e desta forma, procurar deixá-los mais confiáveis quando aplicáveis e implementá-los auxiliados por outras tecnologias, na tentativa de se tornar mais assertivos em termos de precisão e de outras métricas, como por exemplo: falsos positivos e verdadeiros positivos.

Palavras-chave: Visão computacional, Detecção de objetos, OpenCV, Cidades inteligentes, Algoritmos de subtração de fundo.

ABSTRACT

The growth of the population and the fleet of vehicles all over the world, particularly in the big cities, have caused complications in terms of traffic. Solutions for reducing congestion and improving vehicle trafficability require effort and time, which is expensive and ineffective in the medium and long term considering Brazilian municipal administrations. This situation stems mainly from the need for adequate methodologies for efficient monitoring and data collection that public managers can analyze to serve as a basis for road and traffic infrastructure policies. Intelligent traffic monitoring is a new branch of intelligent transport systems that focuses on improving urban traffic conditions. The main objective of systems based on intelligent monitoring is to improve the traffic system, reduce vehicle fluidity problems and improve urban mobility. Many Brazilian cities face congestion due to the inefficient configuration of traffic lights, mainly based on fixed-cycle protocols. Within this context, there is a need to improve and automate the existing traffic light system. A mixed technique of artificial intelligence and computer vision may be desirable to develop a reliable and scalable traffic system that can help solve these problems. It first works in computer vision date back to the late 1970s, when computers could already process large data sets, such as images. In-depth studies in this area begin, and the field of computer vision can be characterized as immature and diverse, even after there are recognized works. Within this context, this dissertation focused on using computer vision technology to build an efficient and resourceful system for synchronous and automated traffic analysis. The performance of different algorithms and systems that apply computer vision to identify and track objects in urban traffic is discussed, detecting pedestrians and vehicles. With this, determining which algorithms and systems would best fit a more complex system based on IoT devices or even autonomous vehicles, which use computer vision systems. Somebody, Systematic Literature Review (SLR) were carried out in articles published from 01/01/1970 to 01/31/2023. As BGS is a technique used in computer vision to detect moving objects in a sequence of images obtained from a video, for example, it was found that relying only on BGS without any other processing or machine learning (ML) support or deep learning (DL), are not accurate enough for direct application in traffic light systems or autonomous monitoring. However, with the constant evolution of specialized software and hardware, the joint application of these solutions is promising. As the results of this work demonstrate, there is a need to improve the individual filters and, in this way, try to make them more reliable when applicable and implement them aided by other technologies, in an attempt to become more assertive in terms of precision and other metrics, such as false positives and true positives.

Keywords: Computer vision, Object detection, OpenCV, Smart cities, Background subtraction algorithms.

LISTA DE FIGURAS

Figura 1 - Imagem preto e branco representada em uma matriz de inteiros	62
Figura 2 - Sistema de coordenadas de imagens - Linha e Coluna.....	63
Figura 3 - Modelo RGB.....	64
Figura 4 - Sistema RGB de cores	64
Figura 5 - Modelo HSI	65
Figura 6 - Exemplos de oclusão	67
Figura 7 - Áreas que formam a Visão Computacional	70
Figura 8 - Níveis de processamento de imagens	70
Figura 9 - Filtro Haar.....	73
Figura 10 - Cubo de cores RGB	82
Figura 11 - Cubo YCrCb	82
Figura 12 - kNN segundo OpenCV.org	83
Figura 13 - Haar.....	91
Figura 14 - AdaBoost.....	92
Figura 15 - HAAR CASCADES por Viola e Jones.....	93
Figura 16 - HAAR CASCADES	94
Figura 17 - Ilustração de como a imagem integral funciona	95
Figura 18 - Demonstração: (a) Quadro de vídeo original, (b) Primeiro plano detectado, (c) Operação Morfológica, (d) Veículos Detectados.....	97
Figura 19 – Sistema de detecção de veículos proposto por PEÑA-GONZÁLEZ e NUÑO-MAGANDA, 2014	101
Figura 20 – Layout proposto por KANUNGO, SHARMA e SINGLA, 2014	104
Figura 21 - Sistema proposto por (ZINCHENKO, KONDRATENKO, et al., 2020)	108
Figura 22 - Procedimentos Realizados	115
Figura 23 - Interface criada para auxiliar o leitor.....	141
Figura 24 – Óbitos por causas externas conforme DATASUS – Brasil	276

LISTA DE IMAGENS

Imagem 1 - Zonas de detecção configuradas na tela do monitor de vídeo.....	53
Imagem 2 - Espaços de cores: Cores, Tons de Cinza, L*a*b e HSV	66
Imagem 3 - Sistema de visão computacional na detecção de veículos, identificando matrículas e fornecendo informações sobre a marca, cor, tipo e posição do veículo.	77
Imagem 4 - Exemplos de quadros de vídeo de cada uma das 11 categorias no conjunto de dados CDNET 2014.....	96
Imagem 5 - Veículos em condições ambientais urbanas e rodoviárias.....	106
Imagem 6 – Quadro (frame) inicial do vídeo "Traffic_3.mp4"	124
Imagem 7 – Frame 0 - Status inicial do vídeo Traffic_3.mp4.....	205
Imagem 8 – Frame 20 Traffic_3.mp4	206
Imagem 9 – Frame 80 Traffic_3.mp4	206
Imagem 10 – Frame 133 Traffic_3.mp4	206
Imagem 11 - Frame 193 Traffic_3.mp4.....	206
Imagem 12 - Frame 262 Traffic_3.mp4.....	206
Imagem 13 - Frame 307 Traffic_3.mp4.....	206
Imagem 14 - Frame 356 Traffic_3.mp4.....	206
Imagem 15 - Frame 374 Traffic_3.mp4.....	206
Imagem 16 - Frame 408 Traffic_3.mp4.....	207
Imagem 17 - Frame 673 Traffic_3.mp4.....	207
Imagem 18 - Frame 490 Traffic_3.mp4.....	207
Imagem 19 - Frame 710 Traffic_3.mp4.....	207
Imagem 20 - Frame 527 Traffic_3.mp4.....	207
Imagem 21 - Frame 790 Traffic_3.mp4.....	207
Imagem 22 - Frame 616 Traffic_3.mp4.....	207
Imagem 23 - Frame 838 Traffic_3.mp4.....	207
Imagem 24 - Frame 933 Traffic_3.mp4.....	208
Imagem 25 - Frame 950 Traffic_3.mp4.....	208
Imagem 26 - Frame 1098 Traffic_3.mp4.....	208
Imagem 27 - Frame 1130 Traffic_3.mp4.....	208
Imagem 28 - Frame 1165 Traffic_3.mp4.....	208

Imagem 29 - Frame 1190 Traffic_3.mp4	208
Imagem 30 - Frame 1235 Traffic_3.mp4	208
Imagem 31 - Frame 1264 Traffic_3.mp4	208
Imagem 32 - Frame 1305 Traffic_3.mp4	209
Imagem 33 - Frame 1458 Traffic_3.mp4	209
Imagem 34 - Frame 1478 Traffic_3.mp4	209
Imagem 35 - Frame 1570 Traffic_3.mp4	209
Imagem 36 - Frame 1678 Traffic_3.mp4	209
Imagem 37 - Frame 1705 Traffic_3.mp4	209
Imagem 38 - Frame 1768 Traffic_3.mp4	209
Imagem 39 - Frame 1851 Traffic_3.mp4	209
Imagem 40 - Frame 1900 Traffic_3.mp4	210
Imagem 41 - Frame 1920 Traffic_3.mp4	210
Imagem 42 - Frame 1950 Traffic_3.mp4	210
Imagem 43 - Frame 1996 Traffic_3.mp4	210
Imagem 44 - Frame 2044 Traffic_3.mp4	210
Imagem 45 - Frame 2084 Traffic_3.mp4	210
Imagem 46 - Frame 2113 Traffic_3.mp4	210
Imagem 47 - Frame 2210 Traffic_3.mp4	210
Imagem 48 - Frame 2265 Traffic_3.mp4	211
Imagem 49 - Frame 2321 Traffic_3.mp4	211
Imagem 50 - Frame 2351 Traffic_3.mp4	211
Imagem 51 - Frame 2407 Traffic_3.mp4	211
Imagem 52 - Frame 2470 Traffic_3.mp4	211
Imagem 53 - Frame 2517 Traffic_3.mp4	211
Imagem 54 - Frame 2560 Traffic_3.mp4	211
Imagem 55 - Frame 2620 Traffic_3.mp4	211
Imagem 56 - Frame 2654 Traffic_3.mp4	212
Imagem 57 - Frame 2718 Traffic_3.mp4	212
Imagem 58 - Frame 2750 Traffic_3.mp4	212
Imagem 59 - Frame 2873 Traffic_3.mp4	212
Imagem 60 - Frame 2905 Traffic_3.mp4	212

Imagem 61 - Frame 2943 Traffic_3.mp4.....	212
Imagem 62 - Frame 3020 Traffic_3.mp4.....	212
Imagem 63 - Frame 3118 Traffic_3.mp4.....	212
Imagem 64 - Frame 3162 Traffic_3.mp4.....	213
Imagem 65 - Frame 3233 Traffic_3.mp4.....	213
Imagem 66 - Frame 3278 Traffic_3.mp4.....	213
Imagem 67 - Frame 3316 Traffic_3.mp4.....	213
Imagem 68 - Frame 3338 Traffic_3.mp4.....	213
Imagem 69 - Frame 3380 Traffic_3.mp4.....	213
Imagem 70 - Frame 3440 Traffic_3.mp4.....	213
Imagem 71 - Frame 3490 Traffic_3.mp4.....	213
Imagem 72 - Frame 3575 Traffic_3.mp4.....	214
Imagem 73 - Frame 3660 Traffic_3.mp4.....	214
Imagem 74 - Frame 3727 Traffic_3.mp4.....	214
Imagem 75 - Frame 3819 Traffic_3.mp4.....	214
Imagem 76 - Frame 3851 Traffic_3.mp4.....	214
Imagem 77 - Frame 3892 Traffic_3.mp4.....	214
Imagem 78 - Frame 3924 Traffic_3.mp4.....	214
Imagem 79 - Frame 3950 Traffic_3.mp4.....	214
Imagem 80 - Frame 3998 Traffic_3.mp4.....	215
Imagem 81 - Frame 4080 Traffic_3.mp4.....	215
Imagem 82 - Frame 4177 Traffic_3.mp4.....	215
Imagem 83 - Frame 4188 Traffic_3.mp4.....	215
Imagem 84 - Frame 4255 Traffic_3.mp4.....	215
Imagem 85 - Frame 4314 Traffic_3.mp4.....	215
Imagem 86 - Frame 4352 Traffic_3.mp4.....	215
Imagem 87 - Frame 4455 Traffic_3.mp4.....	215
Imagem 88 - Frame 4480 Traffic_3.mp4.....	216
Imagem 89 - Frame 4563 Traffic_3.mp4.....	216
Imagem 90 - Frame 4688 Traffic_3.mp4.....	216
Imagem 91 - Frame 4730 Traffic_3.mp4.....	216
Imagem 92 - Frame 4790 Traffic_3.mp4.....	216

Imagem 93 - Frame 4832 Traffic_3.mp4	216
Imagem 94 - Frame 4848 Traffic_3.mp4	216
Imagem 95 - Frame 4871 Traffic_3.mp4	216
Imagem 96 - Frame 4968 Traffic_3.mp4	217
Imagem 97 - Frame 4982 Traffic_3.mp4	217
Imagem 98 - Frame 5245 Traffic_3.mp4	217
Imagem 99 - Frame 5292 Traffic_3.mp4	217
Imagem 100 - Frame 5415 Traffic_3.mp4	217
Imagem 101 - Frame 5525 Traffic_3.mp4	217
Imagem 102 - Frame 5572 Traffic_3.mp4	217
Imagem 103 - Frame 5603 Traffic_3.mp4	217
Imagem 104 - Frame 5672 Traffic_3.mp4	218
Imagem 105 - Frame 5735 Traffic_3.mp4	218
Imagem 106 - Frame 5790 Traffic_3.mp4	218
Imagem 107 - Frame 5822 Traffic_3.mp4	218
Imagem 108 - Frame 5850 Traffic_3.mp4	218
Imagem 109 - Frame 5965 Traffic_3.mp4	218
Imagem 110 - Frame 6090 Traffic_3.mp4	218
Imagem 111 - Frame 6121 Traffic_3.mp4.....	218
Imagem 112 - Frame 6187 Traffic_3.mp4	219
Imagem 113 - Frame 6257 Traffic_3.mp4	219
Imagem 114 - Frame 6295 Traffic_3.mp4	219
Imagem 115 - Frame 6370 Traffic_3.mp4	219
Imagem 116 - Frame 6410 Traffic_3.mp4	219
Imagem 117 - Frame 6438 Traffic_3.mp4	219
Imagem 118 - Frame 6480 Traffic_3.mp4	219
Imagem 119 - Frame 6533 Traffic_3.mp4	219
Imagem 120 - Frame 6585 Traffic_3.mp4	220
Imagem 121 - Frame 6615 Traffic_3.mp4	220
Imagem 122 - Frame 6665 Traffic_3.mp4	220
Imagem 123 - Frame 6727 Traffic_3.mp4	220
Imagem 124 - Frame 6768 Traffic_3.mp4	220

Imagem 125 - Frame 6805 Traffic_3.mp4.....	220
Imagem 126 - Frame 6832 Traffic_3.mp4.....	220
Imagem 127 - Frame 6910 Traffic_3.mp4.....	220
Imagem 128 - Frame 6936 Traffic_3.mp4.....	221
Imagem 129 - Frame 7030 Traffic_3.mp4.....	221
Imagem 130 - Frame 7151 Traffic_3.mp4.....	221
Imagem 131 - Frame 7254 Traffic_3.mp4.....	221
Imagem 132 - Frame 7330 Traffic_3.mp4.....	221
Imagem 133 - Frame 7360 Traffic_3.mp4.....	221
Imagem 134 - Frame 7400 Traffic_3.mp4.....	221
Imagem 135 - Frame 7413 Traffic_3.mp4.....	221
Imagem 136 - Frame 7485 Traffic_3.mp4.....	222
Imagem 137 - Frame 7556 Traffic_3.mp4.....	222
Imagem 138 - Frame 7574 Traffic_3.mp4.....	222
Imagem 139 - Frame 7641 Traffic_3.mp4.....	222
Imagem 140 - Frame 7696 Traffic_3.mp4.....	222
Imagem 141 - Frame 7777 Traffic_3.mp4.....	222
Imagem 142 - Frame 7835 Traffic_3.mp4.....	222
Imagem 143 - Frame 7896 Traffic_3.mp4.....	222
Imagem 144 - Frame 7960 Traffic_3.mp4.....	223
Imagem 145 - Frame 8135 Traffic_3.mp4.....	223
Imagem 146 - Frame 8157 Traffic_3.mp4.....	223
Imagem 147 - Frame 8224 Traffic_3.mp4.....	223
Imagem 148 - Frame 8257 Traffic_3.mp4.....	223
Imagem 149 - Frame 8390 Traffic_3.mp4.....	223
Imagem 150 - Frame 8553 Traffic_3.mp4.....	223
Imagem 151 - Frame 8650 Traffic_3.mp4.....	223
Imagem 152 - Frame 8700 Traffic_3.mp4.....	224
Imagem 153 - Frame 8815 Traffic_3.mp4.....	224
Imagem 154 - Frame 8870 Traffic_3.mp4.....	224
Imagem 155 - Frame 8940 Traffic_3.mp4.....	224
Imagem 156 - Frame 9045 Traffic_3.mp4.....	224

Imagem 157 - Frame 9100 Traffic_3.mp4	224
Imagem 158 - Frame 9140 Traffic_3.mp4	224
Imagem 159 - Frame 9180 Traffic_3.mp4	224
Imagem 160 - Frame 9225 Traffic_3.mp4	225
Imagem 161 - Frame 9285 Traffic_3.mp4	225
Imagem 162 - Frame 9346 Traffic_3.mp4	225
Imagem 163 - Frame 9435 Traffic_3.mp4	225
Imagem 164 - Frame 9480 Traffic_3.mp4	225
Imagem 165 - Frame 9570 Traffic_3.mp4	225
Imagem 166 - Frame 9750 Traffic_3.mp4	225
Imagem 167 - Frame 9816 Traffic_3.mp4	225
Imagem 168 - Frame 9857 Traffic_3.mp4	226
Imagem 169 - Frame 9906 Traffic_3.mp4	226
Imagem 170 - Frame 9930 Traffic_3.mp4	226
Imagem 171 - Frame 10010 Traffic_3.mp4	226
Imagem 172 - Frame 10047 Traffic_3.mp4	226
Imagem 173 - Frame 10080 Traffic_3.mp4	226
Imagem 174 - Frame 10123 Traffic_3.mp4	226
Imagem 175 - Frame 10180 Traffic_3.mp4	226
Imagem 176 - Frame 10288 Traffic_3.mp4	227
Imagem 177 - Frame 10337 Traffic_3.mp4	227
Imagem 178 - Frame 10390 Traffic_3.mp4	227
Imagem 179 - Frame 10547 Traffic_3.mp4	227
Imagem 180 - Frame 10590 Traffic_3.mp4	227
Imagem 181 - Frame 10614 Traffic_3.mp4	227
Imagem 182 - Frame 10675 Traffic_3.mp4	227
Imagem 183 - Frame 10730 Traffic_3.mp4	227
Imagem 184 - Frame 10820 Traffic_3.mp4	228
Imagem 185 - Frame 10855 Traffic_3.mp4	228
Imagem 186 - Frame 10980 Traffic_3.mp4	228
Imagem 187 - Frame 11038 Traffic_3.mp4	228
Imagem 188 - Frame 11090 Traffic_3.mp4	228

Imagem 189 - Frame 11160 Traffic_3.mp4	228
Imagem 190 - Frame 11190 Traffic_3.mp4	228
Imagem 191 - Frame 11230 Traffic_3.mp4	228
Imagem 192 - Frame 11300 Traffic_3.mp4	229
Imagem 193 - Frame 11339 Traffic_3.mp4	229
Imagem 194 - Frame 11430 Traffic_3.mp4	229
Imagem 195 - Frame 11450 Traffic_3.mp4	229
Imagem 196 - Frame 11481 Traffic_3.mp4	229
Imagem 197 - Frame 11510 Traffic_3.mp4	229
Imagem 198 - Frame 11600 Traffic_3.mp4	229
Imagem 199 - Frame 11630 Traffic_3.mp4	229
Imagem 200 - Frame 11700 Traffic_3.mp4	230
Imagem 201 - Frame 11720 Traffic_3.mp4	230
Imagem 202 - Frame 11770 Traffic_3.mp4	230
Imagem 203 - Frame 11915 Traffic_3.mp4	230
Imagem 204 - Frame 11964 Traffic_3.mp4	230
Imagem 205 - Frame 12023 Traffic_3.mp4	230
Imagem 206 - Frame 12070 Traffic_3.mp4	230
Imagem 207 - Frame 12115 Traffic_3.mp4	230
Imagem 208 - Frame 12230 Traffic_3.mp4	231
Imagem 209 - Frame 12285 Traffic_3.mp4	231
Imagem 210 - Frame 12360 Traffic_3.mp4	231
Imagem 211 - Frame 12450 Traffic_3.mp4	231
Imagem 212 - Frame 12490 Traffic_3.mp4	231
Imagem 213 - Frame 12528 Traffic_3.mp4	231
Imagem 214 - Frame 12560 Traffic_3.mp4	231
Imagem 215 - Frame 12607 Traffic_3.mp4	231
Imagem 216 - Frame 12655 Traffic_3.mp4	232
Imagem 217 - Frame 12685 Traffic_3.mp4	232
Imagem 218 - Frame 12800 Traffic_3.mp4	232
Imagem 219 - Frame 12850 Traffic_3.mp4	232
Imagem 220 - Frame 12860 Traffic_3.mp4	232

Imagem 221 - Frame 12894 Traffic_3.mp4	232
Imagem 222 - Frame 12920 Traffic_3.mp4	232
Imagem 223 - Frame 12970 Traffic_3.mp4	232
Imagem 224 - Frame 12980 Traffic_3.mp4	233
Imagem 225 - Frame 13040 Traffic_3.mp4	233
Imagem 226 - Frame 13148 Traffic_3.mp4	233
Imagem 227 - Frame 13196 Traffic_3.mp4	233
Imagem 228 - Frame 13246 Traffic_3.mp4	233
Imagem 229 - Frame 13365 Traffic_3.mp4	233
Imagem 230 - Frame 13416 Traffic_3.mp4	233
Imagem 231 - Frame 13445 Traffic_3.mp4	233
Imagem 232 - Frame 13464 Traffic_3.mp4	234
Imagem 233 - Frame 13500 Traffic_3.mp4	234
Imagem 234 - Frame 13540 Traffic_3.mp4	234
Imagem 235 - Frame 13642 Traffic_3.mp4	234
Imagem 236 - Frame 13663 Traffic_3.mp4	234
Imagem 237 - Frame 13695 Traffic_3.mp4	234
Imagem 238 - Frame 13740 Traffic_3.mp4	234
Imagem 239 - Frame 13752 Traffic_3.mp4	234
Imagem 240 - Frame 13840 Traffic_3.mp4	235
Imagem 241 - Frame 13910 Traffic_3.mp4	235
Imagem 242 - Frame 13948 Traffic_3.mp4	235
Imagem 243 - Frame 13964 Traffic_3.mp4	235
Imagem 244 - Frame 14106 Traffic_3.mp4	235
Imagem 245 - Frame 14165 Traffic_3.mp4	235
Imagem 246 - Frame 14226 Traffic_3.mp4	235
Imagem 247 - Frame 14273 Traffic_3.mp4	235
Imagem 248 - Frame 14452 Traffic_3.mp4	236
Imagem 249 - Frame 14614 Traffic_3.mp4	236
Imagem 250 - Frame 14648 Traffic_3.mp4	236
Imagem 251 - Frame 14695 Traffic_3.mp4	236
Imagem 252 - Frame 14720 Traffic_3.mp4	236

Imagem 253 - Frame 14754 Traffic_3.mp4.....	236
Imagem 254 - Frame 14866 Traffic_3.mp4.....	236
Imagem 255 - Frame 14920 Traffic_3.mp4.....	236
Imagem 256 - Frame 14952 Traffic_3.mp4.....	237
Imagem 257 - Frame 14972 Traffic_3.mp4.....	237
Imagem 258 - Frame 15016 Traffic_3.mp4.....	237
Imagem 259 - Frame 15066 Traffic_3.mp4.....	237
Imagem 260 - Frame 15104 Traffic_3.mp4.....	237
Imagem 261 - Frame 15134 Traffic_3.mp4.....	237
Imagem 262 - Frame 15168 Traffic_3.mp4.....	237
Imagem 263 - Frame 15208 Traffic_3.mp4.....	237
Imagem 264 - Frame 15280 Traffic_3.mp4.....	238
Imagem 265 - Frame 15316 Traffic_3.mp4.....	238
Imagem 266 - Frame 15386 Traffic_3.mp4.....	238
Imagem 267 - Frame 15408 Traffic_3.mp4.....	238
Imagem 268 - Frame 15488 Traffic_3.mp4.....	238
Imagem 269 - Frame 15560 Traffic_3.mp4.....	238
Imagem 270 - Frame 15690 Traffic_3.mp4.....	238
Imagem 271 - Frame 15808 Traffic_3.mp4.....	238
Imagem 272 - Frame 15910 Traffic_3.mp4.....	239
Imagem 273 - Frame 15926 Traffic_3.mp4.....	239
Imagem 274 - Frame 15952 Traffic_3.mp4.....	239
Imagem 275 - Frame 16036 Traffic_3.mp4.....	239
Imagem 276 - Frame 16136 Traffic_3.mp4.....	239
Imagem 277 - Frame 16248 Traffic_3.mp4.....	239
Imagem 278 - Frame 16338 Traffic_3.mp4.....	239
Imagem 279 - Frame 16354 Traffic_3.mp4.....	239
Imagem 280 - Frame 16394 Traffic_3.mp4.....	240
Imagem 281 - Frame 16420 Traffic_3.mp4.....	240
Imagem 282 - Frame 16436 Traffic_3.mp4.....	240
Imagem 283 - Frame 16460 Traffic_3.mp4.....	240
Imagem 284 - Frame 16530 Traffic_3.mp4.....	240

Imagem 285 - Frame 16628 Traffic_3.mp4	240
Imagem 286 - Frame 16694 Traffic_3.mp4	240
Imagem 287 - Frame 16758 Traffic_3.mp4	240
Imagem 288 - Frame 16832 Traffic_3.mp4	241
Imagem 289 - Frame 16852 Traffic_3.mp4	241
Imagem 290 - Frame 16914 Traffic_3.mp4	241
Imagem 291 - Frame 16944 Traffic_3.mp4	241
Imagem 292 - Frame 16990 Traffic_3.mp4	241
Imagem 293 - Frame 17028 Traffic_3.mp4	241
Imagem 294 - Frame 17076 Traffic_3.mp4	241
Imagem 295 - Frame 17182 Traffic_3.mp4	241
Imagem 296 - Frame 17202 Traffic_3.mp4	242
Imagem 297 - Frame 17250 Traffic_3.mp4	242
Imagem 298 - Frame 17282 Traffic_3.mp4	242
Imagem 299 - Frame 17324 Traffic_3.mp4	242
Imagem 300 - Frame 17340 Traffic_3.mp4	242
Imagem 301 - Frame 17354 Traffic_3.mp4	242
Imagem 302 - Frame 17398 Traffic_3.mp4	242
Imagem 303 - Frame 17420 Traffic_3.mp4	242
Imagem 304 - Frame 17436 Traffic_3.mp4	243
Imagem 305 - Frame 17470 Traffic_3.mp4	243
Imagem 306 - Frame 17522 Traffic_3.mp4	243
Imagem 307 - Frame 17546 Traffic_3.mp4	243
Imagem 308 - Frame 17572 Traffic_3.mp4	243
Imagem 309 - Frame 17628 Traffic_3.mp4	243
Imagem 310 - Frame 17650 Traffic_3.mp4	243
Imagem 311 - Frame 17678 Traffic_3.mp4	243
Imagem 312 - Frame 17788 Traffic_3.mp4	244
Imagem 313 - Frame 17834 Traffic_3.mp4	244
Imagem 314 - Frame 17916 Traffic_3.mp4	244
Imagem 315 - Frame 18024 Traffic_3.mp4	244
Imagem 316 - Frame 18112 Traffic_3.mp4	244

Imagem 317 - Frame 18210 Traffic_3.mp4.....	244
Imagem 318 - Frame 18224 Traffic_3.mp4.....	244
Imagem 319 - Frame 18272 Traffic_3.mp4.....	244
Imagem 320 - Frame 18312 Traffic_3.mp4.....	245
Imagem 321 - Frame 18376 Traffic_3.mp4.....	245
Imagem 322 - Frame 18422 Traffic_3.mp4.....	245
Imagem 323 - Frame 18495 Traffic_3.mp4.....	245
Imagem 324 - Frame 18538 Traffic_3.mp4.....	245
Imagem 325 - Frame 18590 Traffic_3.mp4.....	245
Imagem 326 - Frame 18674 Traffic_3.mp4.....	245
Imagem 327 - Frame 18714 Traffic_3.mp4.....	245
Imagem 328 - Frame 18762 Traffic_3.mp4.....	246
Imagem 329 - Frame 18818 Traffic_3.mp4.....	246
Imagem 330 - Frame 18846 Traffic_3.mp4.....	246
Imagem 331 - Frame 18870 Traffic_3.mp4.....	246
Imagem 332 - Frame 18904 Traffic_3.mp4.....	246
Imagem 333 - Frame 18930 Traffic_3.mp4.....	246
Imagem 334 - Frame 18992 Traffic_3.mp4.....	246
Imagem 335 - Frame 19054 Traffic_3.mp4.....	246
Imagem 336 - Frame 19092 Traffic_3.mp4.....	247
Imagem 337 - Frame 19184 Traffic_3.mp4.....	247
Imagem 338 - Frame 19296 Traffic_3.mp4.....	247
Imagem 339 - Frame 19326 Traffic_3.mp4.....	247
Imagem 340 - Frame 19338 Traffic_3.mp4.....	247
Imagem 341 - Frame 19350 Traffic_3.mp4.....	247
Imagem 342 - Frame 19370 Traffic_3.mp4.....	247
Imagem 343 - Frame 19432 Traffic_3.mp4.....	247
Imagem 344 - Frame 19444 Traffic_3.mp4.....	248
Imagem 345 - Frame 19476 Traffic_3.mp4.....	248
Imagem 346 - Frame 19502 Traffic_3.mp4.....	248
Imagem 347 - Frame 19536 Traffic_3.mp4.....	248
Imagem 348 - Frame 19560 Traffic_3.mp4.....	248

Imagem 349 - Frame 19632 Traffic_3.mp4	248
Imagem 350 - Frame 19732 Traffic_3.mp4	248
Imagem 351 - Frame 19764 Traffic_3.mp4	248
Imagem 352 - Frame 19786 Traffic_3.mp4	249
Imagem 353 - Frame 19828 Traffic_3.mp4	249
Imagem 354 - Frame 19842 Traffic_3.mp4	249
Imagem 355 - Frame 19892 Traffic_3.mp4	249
Imagem 356 - Frame 19910 Traffic_3.mp4	249
Imagem 357 - Frame 19984 Traffic_3.mp4	249
Imagem 358 - Frame 20048 Traffic_3.mp4	249
Imagem 359 - Frame 20072 Traffic_3.mp4	249
Imagem 360 - Frame 20162 Traffic_3.mp4	250
Imagem 361 - Frame 20208 Traffic_3.mp4	250
Imagem 362 - Frame 20312 Traffic_3.mp4	250
Imagem 363 - Frame 20334 Traffic_3.mp4	250
Imagem 364 - Frame 20360 Traffic_3.mp4	250
Imagem 365 - Frame 20408 Traffic_3.mp4	250
Imagem 366 - Frame 20460 Traffic_3.mp4	250
Imagem 367 - Frame 20528 Traffic_3.mp4	250
Imagem 368 - Frame 20548 Traffic_3.mp4	251
Imagem 369 - Frame 20570 Traffic_3.mp4	251
Imagem 370 - Frame 20614 Traffic_3.mp4	251
Imagem 371 - Frame 20640 Traffic_3.mp4	251
Imagem 372 - Frame 20660 Traffic_3.mp4	251
Imagem 373 - Frame 20734 Traffic_3.mp4	251
Imagem 374 - Frame 20747 Traffic_3.mp4	251
Imagem 375 - Frame 20784 Traffic_3.mp4	251
Imagem 376 - Frame 20806 Traffic_3.mp4	252
Imagem 377 - Frame 20842 Traffic_3.mp4	252
Imagem 378 - Frame 20890 Traffic_3.mp4	252
Imagem 379 - Frame 21002 Traffic_3.mp4	252
Imagem 380 - Frame 21080 Traffic_3.mp4	252

Imagem 381 - Frame 21136 Traffic_3.mp4.....	252
Imagem 382 - Frame 21212 Traffic_3.mp4.....	252
Imagem 383 - Frame 21264 Traffic_3.mp4.....	252
Imagem 384 - Frame 21312 Traffic_3.mp4.....	253
Imagem 385 - Frame 21366 Traffic_3.mp4.....	253
Imagem 386 - Frame 21488 Traffic_3.mp4.....	253
Imagem 387 - Frame 21514 Traffic_3.mp4.....	253
Imagem 388 - Frame 21528 Traffic_3.mp4.....	253
Imagem 389 - Frame 21622 Traffic_3.mp4.....	253
Imagem 390 - Frame 21674 Traffic_3.mp4.....	253
Imagem 391 - Frame 21792 Traffic_3.mp4.....	253
Imagem 392 - Frame 21856 Traffic_3.mp4.....	254
Imagem 393 - Frame 21908 Traffic_3.mp4.....	254
Imagem 394 - Frame 21932 Traffic_3.mp4.....	254
Imagem 395 - Frame 21966 Traffic_3.mp4.....	254
Imagem 396 - Frame 21982 Traffic_3.mp4.....	254
Imagem 397 - Frame 22026 Traffic_3.mp4.....	254
Imagem 398 - Frame 22080 Traffic_3.mp4.....	254
Imagem 399 - Frame 22118 Traffic_3.mp4.....	254
Imagem 400 - Frame 22172 Traffic_3.mp4.....	255
Imagem 401 - Frame 22272 Traffic_3.mp4.....	255
Imagem 402 - Frame 22302 Traffic_3.mp4.....	255
Imagem 403 - Frame 22336 Traffic_3.mp4.....	255
Imagem 404 - Frame 22356 Traffic_3.mp4.....	255
Imagem 405 - Frame 22390 Traffic_3.mp4.....	255
Imagem 406 - Frame 22472 Traffic_3.mp4.....	255
Imagem 407 - Frame 22526 Traffic_3.mp4.....	255
Imagem 408 - Frame 22622 Traffic_3.mp4.....	256
Imagem 409 - Frame 22734 Traffic_3.mp4.....	256
Imagem 410 - Frame 22778 Traffic_3.mp4.....	256
Imagem 411 - Frame 22830 Traffic_3.mp4.....	256
Imagem 412 - Frame 22866 Traffic_3.mp4.....	256

Imagem 413 - Frame 22962 Traffic_3.mp4	256
Imagem 414 - Frame 23070 Traffic_3.mp4	256
Imagem 415 - Frame 23178 Traffic_3.mp4	256
Imagem 416 - Frame 23220 Traffic_3.mp4	257
Imagem 417 - Frame 23236 Traffic_3.mp4	257
Imagem 418 - Frame 23284 Traffic_3.mp4	257
Imagem 419 - Frame 23330 Traffic_3.mp4	257
Imagem 420 - Frame 23362 Traffic_3.mp4	257
Imagem 421 - Frame 23398 Traffic_3.mp4	257
Imagem 422 - Frame 23434 Traffic_3.mp4	257
Imagem 423 - Frame 23502 Traffic_3.mp4	257
Imagem 424 - Frame 23530 Traffic_3.mp4	258
Imagem 425 - Frame 23604 Traffic_3.mp4	258
Imagem 426 - Frame 23650 Traffic_3.mp4	258
Imagem 427 - Frame 23668 Traffic_3.mp4	258
Imagem 428 - Frame 23770 Traffic_3.mp4	258
Imagem 429 - Frame 23864 Traffic_3.mp4	258
Imagem 430 - Frame 23938 Traffic_3.mp4	258
Imagem 431 - Frame 24078 Traffic_3.mp4	258
Imagem 432 - Frame 24150 Traffic_3.mp4	259
Imagem 433 - Frame 24172 Traffic_3.mp4	259
Imagem 434 - Frame 24212 Traffic_3.mp4	259
Imagem 435 - Frame 24238 Traffic_3.mp4	259
Imagem 436 - Frame 24272 Traffic_3.mp4	259
Imagem 437 - Frame 24334 Traffic_3.mp4	259
Imagem 438 - Frame 24362 Traffic_3.mp4	259
Imagem 439 - Frame 24394 Traffic_3.mp4	259
Imagem 440 - Frame 24430 Traffic_3.mp4	260
Imagem 441 - Frame 24454 Traffic_3.mp4	260
Imagem 442 - Frame 24500 Traffic_3.mp4	260
Imagem 443 - Frame 24544 Traffic_3.mp4	260
Imagem 444 - Frame 24600 Traffic_3.mp4	260

Imagem 445 - Frame 24632 Traffic_3.mp4.....	260
Imagem 446 - Frame 24676 Traffic_3.mp4.....	260
Imagem 447 - Frame 24708 Traffic_3.mp4.....	260
Imagem 448 - Frame 24768 Traffic_3.mp4.....	261
Imagem 449 - Frame 24824 Traffic_3.mp4.....	261
Imagem 450 - Frame 24852 Traffic_3.mp4.....	261
Imagem 451 - Frame 24878 Traffic_3.mp4.....	261
Imagem 452 - Frame 24928 Traffic_3.mp4.....	261
Imagem 453 - Frame 24970 Traffic_3.mp4.....	261
Imagem 454 - Frame 25110 Traffic_3.mp4.....	261
Imagem 455 - Frame 25138 Traffic_3.mp4.....	261
Imagem 456 - Frame 25178 Traffic_3.mp4.....	262
Imagem 457 - Frame 25189 Traffic_3.mp4.....	262
Imagem 458 - Carro_Descendo_0141.png GMG.....	264
Imagem 459 - Carro_Descendo_0171.png GMG.....	264
Imagem 460 - Carro_Descendo_0259.png GMG.....	264
Imagem 461 - Carro_Subindo_0295.png GMG.....	264
Imagem 462 - Carro_Descendo_0321.png GMG.....	264
Imagem 463 - Caminhao_Subindo_0317.png GMG.....	264
Imagem 464 - Caminhao_Subindo_0434.png GMG.....	264
Imagem 465 - Caminhao_Subindo_0550.png GMG.....	264
Imagem 466 - Caminhao_Descendo_0916.png GMG.....	264
Imagem 467 - Caminhao_Descendo_1074.png GMG.....	264
Imagem 468 - Carro_Subindo_0047.png MOG.....	265
Imagem 469 - Carro_Descendo_0065.png MOG.....	265
Imagem 470 - Carro_Descendo_0097.png MOG.....	265
Imagem 471 - Carro_Descendo_0142.png MOG.....	265
Imagem 472 - Carro_Descendo_0173.png MOG.....	265
Imagem 473 - Caminhao_Subindo_0315.png MOG.....	265
Imagem 474 - Caminhao_Subindo_0426.png MOG.....	265
Imagem 475 - Caminhao_Subindo_0548.png MOG.....	265
Imagem 476 - Caminhao_Subindo_1204.png MOG.....	265

Imagem 477 - Caminhao_Descendo_1352.png MOG	265
Imagem 478 - Carro_Descendo_0095.png MOG2	266
Imagem 479 - Carro_Descendo_0140.png MOG2	266
Imagem 480 - Carro_Descendo_0172.png MOG2	266
Imagem 481 - Carro_Descendo_0258.png MOG2	266
Imagem 482 - Carro_Subindo_0296.png MOG2	266
Imagem 483 - Caminhao_Descendo_0374.png MOG2	266
Imagem 484 - Caminhao_Descendo_1349.png MOG2	266
Imagem 485 - Caminhao_Descendo_1351.png MOG2	266
Imagem 486 - Caminhao_Subindo_2113.png MOG2	266
Imagem 487 - Caminhao_Descendo_2393.png MOG2	266
Imagem 488 - Carro_Descendo_0049.png KNN	267
Imagem 489 - Carro_Descendo_0088.png KNN	267
Imagem 490 - Carro_Descendo_0132.png KNN	267
Imagem 491 - Carro_Subindo_0126.png KNN	267
Imagem 492 - Carro_Subindo_0246.png KNN	267
Imagem 493 - Caminhao_Descendo_0251.png KNN	267
Imagem 494 - Caminhao_Descendo_0287.png KNN	267
Imagem 495 - Caminhao_Subindo_0342.png KNN	267
Imagem 496 - Caminhao_Subindo_0448.png KNN	267
Imagem 497 - Caminhao_Subindo_0450.png KNN	267
Imagem 498 - Carro_Subindo_0045.png CNT	268
Imagem 499 - Carro_Descendo_0065.png CNT	268
Imagem 500 - Carro_Descendo_0096.png CNT	268
Imagem 501 - Carro_Subindo_0109.png CNT	268
Imagem 502 - Carro_Descendo_0141.png CNT	268
Imagem 503 - Caminhao_Subindo_0316.png CNT	268
Imagem 504 - Caminhao_Subindo_0332.png CNT	268
Imagem 505 - Caminhao_Descendo_1304.png CNT	268
Imagem 506 - Caminhao_Descendo_3267.png CNT	268
Imagem 507 - Caminhao_Subindo_4012.png CNT	268
Imagem 508 - Carro_Descendo_0086.png LSBP	269

Imagem 509 - Carro_Descendo_0132.png LSBP	269
Imagem 510 - Carro_Descendo_0164.png LSBP	269
Imagem 511 - Carro_Subindo_0172.png LSBP	269
Imagem 512 - Carro_Descendo_0250.png LSBP	269
Imagem 513 - Caminhao_Descendo_0287.png LSBP	269
Imagem 514 - Caminhao_Subindo_0322.png LSBP.....	269
Imagem 515 - Caminhao_Descendo_0369.png LSBP	269
Imagem 516 - Caminhao_Subindo_0438.png LSBP.....	269
Imagem 517 - Caminhao_Subindo_0555.png LSBP.....	269
Imagem 518 - Carro_Descendo_0087.png GSOC	271
Imagem 519 - Carro_Subindo_0118.png GSOC	271
Imagem 520 - Carro_Descendo_0132.png GSOC	271
Imagem 521 - Carro_Descendo_0165.png GSOC	271
Imagem 522 - Carro_Descendo_0250.png GSOC	271
Imagem 523 - Caminhao_Subindo_0438.png GSOC	271
Imagem 524 - Caminhao_Subindo_0555.png GSOC	271
Imagem 525 - Caminhao_Subindo_1222.png GSOC	271
Imagem 526 - Caminhao_Descendo_1344.png GSOC	271
Imagem 527 - Caminhao_Subindo_2123.png GSOC	271

LISTA DE GRÁFICOS

Gráfico 1 - Frota brasileira de veículos em circulação 2006-2021	49
Gráfico 2 - Estatísticas brasileiras: mortos em acidentes de trânsito	50
Gráfico 3 - Comparação de técnicas DRTCC.....	99
Gráfico 4 – Contagem de veículos acumulativos.	106
Gráfico 5 – Verdadeiros Positivos	135
Gráfico 6 - Falsos Positivos.....	135
Gráfico 7 - Verdadeiros Positivos - Amostragem	137
Gráfico 8 - QQ - Amostragem	137
Gráfico 9 - Amostragem Falsos Positivos	138
Gráfico 10 - QQ da amostragem de falsos positivos.....	138
Gráfico 11 - Evolução da população da cidade de São Paulo	280
Gráfico 12 - Indicadores da cidade de São Paulo.....	281

LISTA DE FLUXOGRAMAS

Fluxograma 1 - Estrutura da dissertação	57
Fluxograma 2 - Sequência padrão de PADI	59
Fluxograma 3 - Processamento digital de imagem.....	61
Fluxograma 4 – Etapas envolvidas na resolução de congestionamento de tráfego.....	102
Fluxograma 5 - Algoritmo de controle proposto por ALI, MALIK, et al., 2015	103
Fluxograma 6 - Método proposto por (SEENOUVONG, WATCHAREERUETAI, et al., 2016)	105

LISTA DE DIAGRAMAS

Diagrama 1 - Diagrama proposto por COIFMAN, BEYMER, et al., 1998	72
Diagrama 2 - Algoritmos de detecção de veículos na estrada	74
Diagrama 3 - Herança para <i>BackgroundSubtractor</i>	79
Diagrama 4 - Subtração básica do plano de fundo	80
Diagrama 5 - Herança para <i>BackgroundSubtractorKNN</i>	85
Diagrama 6 - Herança para <i>BackgroundSubtractorMOG</i>	86
Diagrama 7 - Herança para <i>BackgroundSubtractorMOG2</i>	87
Diagrama 8 - Herança para <i>BackgroundSubtractorGMG</i>	88
Diagrama 9 - Herança para <i>BackgroundSubtractorCNT</i>	88
Diagrama 10 - Herança para <i>BackgroundSubtractorGSOC</i>	89
Diagrama 11 - Herança para <i>BackgroundSubtractorLSBP</i>	90
Diagrama 12 - Metodologia proposta por Singh & Mittal	98
Diagrama 13 - Sistema proposto por Singh & Mittal.....	98
Diagrama 14 - Procedimentos realizados para análise dos algoritmos	112

LISTA DE QUADROS

Quadro 1 - Comparativo: Segmentação x Classificação x Detecção	71
Quadro 2 - Bases de dados consultadas.....	114
Quadro 3 - Combinações dos termos de busca UCLA	116
Quadro 4 - Combinações dos termos de busca ACM Digital Library.....	117
Quadro 5 - Combinações dos termos de busca ACM Digital Library - OpenCV	118
Quadro 6 - Combinações dos termos de busca ACM Digital Library - YOLO.....	119
Quadro 7 - Combinações dos termos de busca ScienceDirect - BGS	120
Quadro 8 – Filtros aplicados nos algoritmos: GMG, MOG, MOG2, kNN e CNT	127
Quadro 9 – Detalhe da chamada sem filtros nos algoritmos: LSBP e GSOC	129
Quadro 10 – Controle - Teste 01 - GMG.....	144
Quadro 11 - Controle - Teste 01 - MOG.....	145
Quadro 12 - Controle - Teste 01 – MOG2.....	145
Quadro 13 - Controle - Teste 01 - KNN	146
Quadro 14 - Controle - Teste 01 - CNT	147
Quadro 15 - Controle - Teste 01 - LSBP	148
Quadro 16 - Controle - Teste 01 - GSOC	149
Quadro 17 - Métricas de desempenho sugeridas para avaliação em trabalhos futuros	153
Quadro 18 – Código completo do programa de extração de frames de arquivo de vídeo	175
Quadro 19 – Código do programa de testes dos algoritmos testados.....	178
Quadro 20 – Código do programa (módulo obrigatório) – validator.py	187
Quadro 21 - Frames extraídos para o controle da contagem manual dos veículos do vídeo Traffic_3.mp4	206
Quadro 22 - Imagens do controle - Teste 01 – GMG	264
Quadro 23 - Imagens do controle - Teste 01 – MOG	265
Quadro 24 - Imagens do controle - Teste 01 – MOG2	266
Quadro 25 - Imagens do controle - Teste 01 – KNN.....	267
Quadro 26 - Imagens do controle - Teste 01 – CNT.....	268
Quadro 27 - Imagens do controle - Teste 01 – LSBP	269
Quadro 28 - Imagens do controle - Teste 01 – GSOC.....	271

LISTA DE TABELAS

Tabela 1 - Dados relacionadas à São Paulo por Rede Social Brasileira por Cidades Justas e Sustentáveis (a).....	51
Tabela 2 - Dados relacionadas à São Paulo por Rede Social Brasileira por Cidades Justas e Sustentáveis (b)	51
Tabela 3 – Resultados de performance do classificador em fase de teste	102
Tabela 4 - Dados obtidos nos testes comparativos com o vídeo Traffic_3.mp4.....	126
Tabela 5 – Análise descritiva para determinar a normalidade dos dados obtidos	131
Tabela 6 - Comparativo da taxa de acerto dos algoritmos em relação ao controle	132
Tabela 7 - Controle manual da contagem de veículos	133
Tabela 8 - Observações dos resultados de valores p (Shapiro-Wilk) CARROS.....	133
Tabela 9 - Observações dos resultados de valores p (Shapiro-Wilk) CAMINHÕES.....	134
Tabela 10 - Estatística Descritiva - Precisão dos algoritmos da amostragem	135
Tabela 11 - Teste t para uma amostra.....	136
Tabela 12 - Teste à Normalidade (Shapiro-Wilk) Amostragem.....	136
Tabela 13 - Estatística Descritiva da Amostragem de verdadeiros positivos	136
Tabela 14 - Teste t para uma amostra - Falsos Positivos	137
Tabela 15 - Teste à Normalidade (Shapiro-Wilk) - Amostragem - Falsos Positivos.....	137
Tabela 16 - Estatística Descritiva da amostragem de falsos positivos	138
Tabela 17 - ANOVA a um fator (não-paramétrica) Kruskal-Wallis - Amostragem.....	138
Tabela 18 - Comparações múltiplas Dwass-Steel-Critchlow-Fligner - Verdadeiros Positivos	139
Tabela 19 - Comparações múltiplas Dwass-Steel-Critchlow-Fligner - Falsos Positivos	140
Tabela 20 - Controle - Contagem manual dos veículos do vídeo Traffic_3.mp4.....	192
Tabela 21 - Frota de veículos no Brasil (2006-2021)	277
Tabela 22 – Indicadores da cidade de São Paulo (a)	278
Tabela 23 – Indicadores da cidade de São Paulo (b)	279
Tabela 24 - Dados associados da Cidade de São Paulo	280

LISTA DE ABREVIATURAS E SIGLAS

ANN	Artificial Neural Network
apud	Conforme, segundo, citado por
BLOB	Binary Large Object
CFTV	Circuito Fechado de TV
CNN	Convolutional Neural Network
CUDA	Compute Unified Device Architecture
FES	Fuzzy Expert Systems
GPU	Graphics Processing Unit
IA	Inteligência Artificial
ib. ou ibid.	ibidem (da mesma forma)
id.	idem
IDUTC	Intelligence Decision-making system for Urban Traffic-Control
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
IoT	Internet of Things ou Internet das Coisas (em português)
kNN	K-Nearest Neighbors
m	metro
MSL	Mapeamento Sistemático de Literatura
ONU	Organização das Nações Unidas
p.	Página (singular)
PE	Protocolo de Estudo
pp.	Páginas (plural)
QPC	Questão de Pesquisa Central
QP _n	Questão de Pesquisa, onde “ <i>n</i> ” é o número da sua sequência
RSL	Revisão Sistemática de Literatura
STCS	Smart Traffic Control Systems
SVD	Selective Vehicle Detector
SVM	Support vector machine
YOLO	“You Only Look Once” (Ferramenta de visão computacional)

SUMÁRIO

1	INTRODUÇÃO.....	49
1.1	PROBLEMA DE PESQUISA	50
1.2	OBJETIVOS	55
1.2.1	Geral.....	55
1.2.2	Específicos	55
1.3	JUSTIFICATIVA PARA O ESTUDO DO TEMA.....	55
1.4	ESTRUTURA DO TRABALHO.....	56
2	REFERENCIAL TEÓRICO.....	59
2.1	PROCESSAMENTO E ANÁLISE DIGITAL DE IMAGENS	59
2.2	SISTEMA DE COORDENADAS E MANIPULAÇÃO DE PIXELS	62
2.3	SISTEMAS DE CORES.....	63
2.4	CLASSIFICAÇÃO DE IMAGENS.....	66
2.5	OCLUSÃO EM VISÃO COMPUTACIONAL	66
2.6	VISÃO COMPUTACIONAL E INTELIGÊNCIA ARTIFICIAL	68
2.7	OPENCV	76
2.7.1	Subtração básica	79
2.7.2	Descritores (características).....	81
2.7.3	Tipos de Descritores	81
2.7.4	Características de textura	83
2.7.5	Algoritmo: K-Nearest Neighbors (kNN)	83
2.7.6	Algoritmo: Mixture of Gaussians (MoG)	86
2.7.7	Algoritmo: Mixture of Gaussians version 2 (MoG2)	86

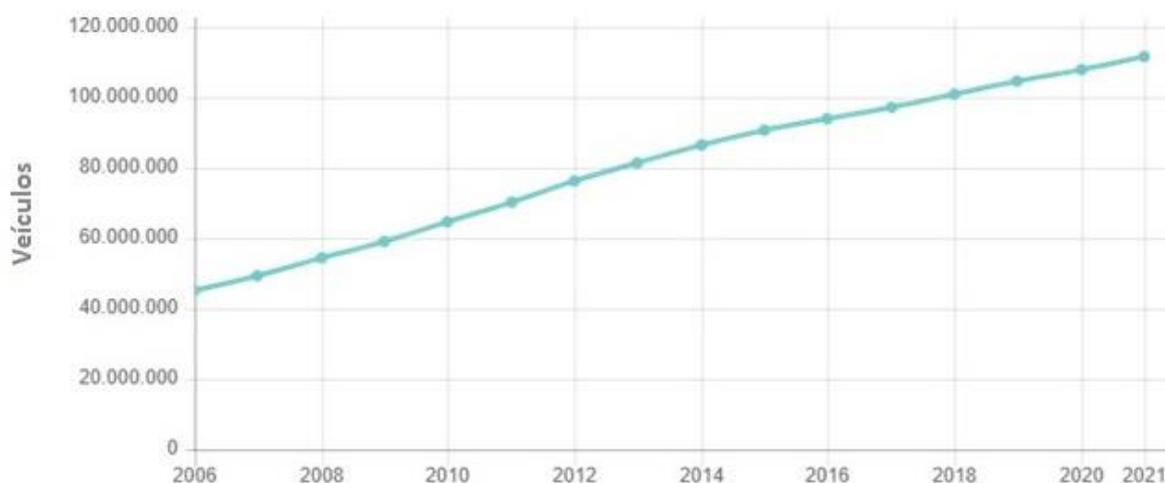
2.7.8	Algoritmo: Godbehere-Matsukawa-Goldberg (GMG).....	87
2.7.9	Algoritmo: CouNT (CNT).....	88
2.7.10	Algoritmo: Google Summer of Code (GSoC).....	89
2.7.11	Algoritmo: Local SVD Binary Pattern (LSBP).....	89
2.7.12	Cascade Classifier, Haar Cascade e Algoritmo AdaBoost	90
2.8	BENCHMARK CDNET 2014	95
2.9	TRABALHOS RELACIONADOS	97
3	MÉTODO E TÉCNICAS DE PESQUISA	111
3.1	CLASSIFICAÇÃO DA PESQUISA	112
3.2	DESCRIÇÃO DO MÉTODO - LITERATURA	113
3.3	DESCRIÇÃO DO MÉTODO – PROCEDIMENTO DOS TESTES	122
3.3.1	Análises estatísticas	129
3.3.1.1	Teste de normalidade (Shapiro-Wilk).....	129
3.3.1.2	Controle manual dos dados	133
3.3.1.3	Análise estatística da amostragem de precisão dos algoritmos.....	134
3.4	DESCRIÇÃO DO SOFTWARE CRIADO PARA REPLICAÇÃO DOS TESTES.	141
4	ANÁLISE E INTERPRETAÇÃO DOS RESULTADOS	142
4.1	CONTROLE DE AMOSTRAGEM DE PRECISÃO DO ALGORITMO GMG.....	143
4.2	CONTROLE DE AMOSTRAGEM DE PRECISÃO DO ALGORITMO MOG.....	144
4.3	CONTROLE DE AMOSTRAGEM DE PRECISÃO DO ALGORITMO MOG2...	145
4.4	CONTROLE DE AMOSTRAGEM DE PRECISÃO DO ALGORITMO KNN	146
4.5	CONTROLE DE AMOSTRAGEM DE PRECISÃO DO ALGORITMO CNT	147
4.6	CONTROLE DE AMOSTRAGEM DE PRECISÃO DO ALGORITMO LSBP	148
4.7	CONTROLE DE AMOSTRAGEM DE PRECISÃO DO ALGORITMO GSOC ...	149
5	CONCLUSÕES	150

5.1	RESPONDENDO AS QUESTÕES DA PESQUISA	151
5.2	CONSIDERAÇÕES FINAIS	152
5.3	TRABALHOS FUTUROS	152
	REFERÊNCIAS	155
	GLOSSÁRIO	169
	APÊNDICE A – CÓDIGO FONTE: OBTENÇÃO DOS FRAMES DE VÍDEOS.....	175
	APÊNDICE B – CÓDIGO FONTE: TESTES DE DESEMPENHO DOS ALGORITMOS BGS.....	177
	APÊNDICE C – EQUIPAMENTOS UTILIZADOS	189
	APÊNDICE D – CONTROLE - VÍDEO TRAFFIC_3.MP4.....	191
	APÊNDICE E – FRAMES EXTRAÍDOS DO VÍDEO TRAFFIC_3.MP4	205
	APÊNDICE F – IMAGENS DE AMOSTRAGEM DOS ALGORITMOS QUANTO A PRECISÃO	263
	APÊNDICE G – CESSÃO DE DIREITOS.....	273
	ANEXO A – DADOS SOBRE ACIDENTES DE TRÂNSITO - BRASIL.....	275
	ANEXO B – DADOS SOBRE A FROTA DE VEÍCULOS NO BRASIL.....	277
	ANEXO C – DADOS ASSOCIADOS SOBRE A CIDADE DE SÃO PAULO – SP - BRASIL.....	278

1 INTRODUÇÃO

Com uma frota de veículos em ascensão, conforme demonstra o (BRASIL - IBGE, 2021), a frota brasileira em 2006 era de 45.029.257 veículos, passou a ser em 2021 de 111.446.870 veículos, ou seja, um aumento de 45%, o que acaba incorrendo em alguns problemas de mobilidade nos grandes centros urbanos do país. No Gráfico 1 encontra-se a evolução da frota de 2006 até 2021.

Gráfico 1 - Frota brasileira de veículos em circulação | 2006-2021



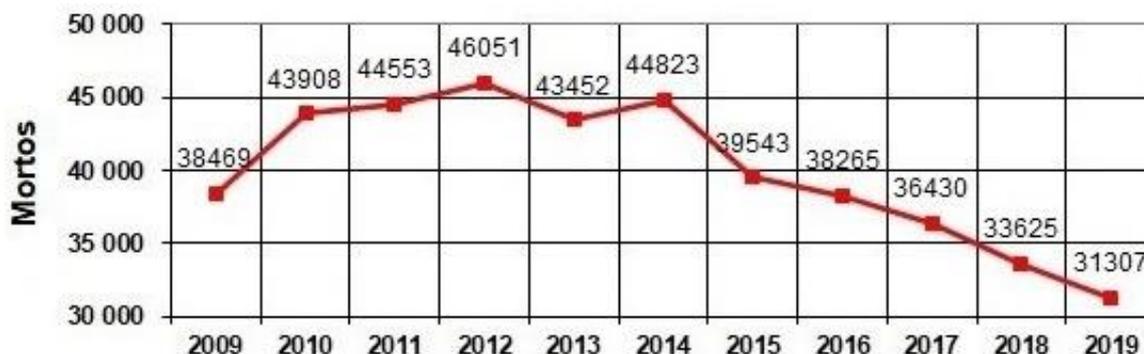
Fonte: (BRASIL - IBGE, 2022)

Alguns dos problemas relacionados ao trânsito urbano de pedestres e veículos podem ser agrupados resumidamente em: acidentes e congestionamentos. Os acidentes de trânsito envolvendo apenas pedestres, segundo o site da *World Health Organization (WHO)* - “*Death on the roads*”, somaram mundialmente 311.614 óbitos em 2018, ou seja, aproximadamente um óbito a cada um minuto e quarenta segundos aproximadamente (WORLD HEALTH ORGANIZATION, 2018).

No Brasil, no mesmo período, foram 38.651 óbitos registrados no “*Global Status Report on Road Safety 2018*”, também elaborado pela Organização Mundial da Saúde (WORLD HEALTH ORGANIZATION, 2018). Ainda, entre 2010 e 2018 houve uma redução de 28,5% (VIAS SEGURAS, 2022).

A publicação “*Road Safety Annual Report 2021*” relatou que ocorreram entre os anos de 2017 e 2019 cerca de 84.850 óbitos de pedestres decorrentes do trânsito. (INTERNATIONAL TRANSPORT FORUM, 2022). O Gráfico 2 apresenta a variação da quantidade de mortes em acidentes de trânsito entre 2009 e 2019.

Gráfico 2 - Estatísticas brasileiras: mortos em acidentes de trânsito



Fonte: (VIAS SEGURAS, 2022)

Mais dados detalhados estão dispostos nos Anexos A, B e C.

1.1 PROBLEMA DE PESQUISA

Embora no Brasil não haja dados oficiais que discriminem exclusivamente acidentes envolvendo pedestres nas proximidades de semáforos, uma consulta ao (BRASIL - DATASUS, 2022) revelou que em 2021 foram registrados 4.484 óbitos envolvendo pedestres, o que equivale aproximadamente a um óbito a cada duas horas.

Por meio de consulta à (REDE SOCIAL BRASILEIRA POR CIDADES JUSTAS E SUSTENTÁVEIS, 2023), foi possível observar os dados que constam nas Tabelas 1 e 2, ambas com informações em relação a cidade de São Paulo.

Em ambas as tabelas, os dados foram filtrados para apresentar apenas a cidade de São Paulo escolhida pelo autor por sua referência geográfica, econômica e proporção de população e frota veicular.

Tabela 1 - Dados relacionadas à São Paulo por Rede Social Brasileira por Cidades Justas e Sustentáveis (a)

Nome do indicador	Período do indicador	Descrição do indicador	Soma
Mortes no trânsito	anual	Número de mortes em acidentes de trânsito, por 100 mil habitantes.	129,15
Mortes com motocicleta		Número de mortes de ocupantes de motocicleta, por 10 mil habitantes.	3,72
Mortes com bicicleta		Número de mortes de ocupantes de bicicleta, por 10 mil habitantes.	0,446
Mortes com automóvel		Número de mortes de ocupantes de automóveis e caminhonetes por 10 mil habitantes	2,08
Acidente com automóvel		Número total de acidentes com vítimas com automóvel por 100 mil habitantes.	166
Acidente com bicicleta		Número total de acidentes com vítimas com bicicleta por 100 mil habitantes.	39,418
Acidente com motocicleta		Número total de acidentes com vítimas com motocicleta por 100 mil habitantes.	285,26

Fontes: Dados filtrados e agrupados com base nos originais em (REDE SOCIAL BRASILEIRA POR CIDADES JUSTAS E SUSTENTÁVEIS, 2023), (REDE SOCIAL BRASILEIRA POR CIDADES JUSTAS E SUSTENTÁVEIS (A), 2022), (REDE SOCIAL BRASILEIRA POR CIDADES JUSTAS E SUSTENTÁVEIS (B), 2022), (REDE SOCIAL BRASILEIRA POR CIDADES JUSTAS E SUSTENTÁVEIS (C), 2022), (REDE SOCIAL BRASILEIRA POR CIDADES JUSTAS E SUSTENTÁVEIS (D), 2022) e (REDE SOCIAL BRASILEIRA POR CIDADES JUSTAS E SUSTENTÁVEIS (E), 2022)

Tabela 2 - Dados relacionadas à São Paulo por Rede Social Brasileira por Cidades Justas e Sustentáveis (b)

Nome do indicador	Descrição do indicador	Média	Soma	Mínimo	Máximo
Congestionamento	Média aritmética mensal (dos dias úteis) dos congestionamentos, em km, nos horários de pico (manhã e tarde)	99,3	298,0	94,0	105,0
Atropelamentos	Número total de atropelamentos por 10 mil habitantes.	2,20	21,97	0,616	3,22
Acidentes de trânsito	Número total de acidentes de trânsito por 10 mil habitantes.	7,77	77,7	2,43	10,7

Fontes: Dados filtrados e agrupados com base nos originais em (REDE SOCIAL BRASILEIRA POR CIDADES JUSTAS E SUSTENTÁVEIS, 2023), (REDE SOCIAL BRASILEIRA POR CIDADES JUSTAS E SUSTENTÁVEIS (E), 2022) e (REDE SOCIAL BRASILEIRA POR CIDADES JUSTAS E SUSTENTÁVEIS (F), 2022)

A cidade de São Paulo já registrou em um único dia, 293km de vias paradas em junho de 2009, porém não é o único exemplo de mega congestionamentos e, pelo mundo, existem exemplos similares, como: Pequim na China, Bethel, Chicago e Nova York nos Estados Unidos, Berlim Ocidental/Oriental na Alemanha), Estrada Lyon-Paris na França, Moscou na Rússia e Tóquio no Japão, conforme apontou a (FORBES, 2017).

Com base apenas as informações anteriores e desconsiderando os números das internações com sobreviventes decorrentes das mesmas causas, e de outros fatores indiretamente impactados por estes dois problemas (acidentes e congestionamentos), pressupõe-se que haja alternativas de propostas de soluções, entre elas, as tecnologias baseadas em visão computacional se destacam.

Um exemplo da utilização de visão computacional no Brasil, é na detecção de veículos em vias públicas, como laço virtual de detecção de tráfego, descrita como um sistema que consiste em sobrepor zonas de detecção em posições adequadas, sobre a imagem da via monitorada e à medida que os veículos percorrem a via cruzando essas zonas, que são ativadas pela mudança do padrão da imagem e resultando então na detecção dos veículos.

Conforme demonstra a Imagem 1 p.53 e descrita no item 3.5.4.3, p.25 do “Manual Brasileiro de Sinalização de Trânsito – Volume V – Sinalização Semafórica”, (BRASIL - CONTRAN | CONSELHO NACIONAL DE TRÂNSITO - MINISTÉRIO DAS CIDADES, 2014).

Imagem 1 - Zonas de detecção configuradas na tela do monitor de vídeo.



Fonte: (BRASIL - CONTRAN | CONSELHO NACIONAL DE TRÂNSITO - MINISTÉRIO DAS CIDADES, 2014)

Como descrito por (PEREZ, 2021), com o crescimento da população, o número de viajantes aumenta constantemente enquanto os recursos fornecidos pelas atuais infraestruturas não são suficientes para atender a demanda, portanto, o controle de tráfego tornou-se um problema sério. O trânsito nas áreas urbanas é geralmente controlado com base em semáforos de ciclo fixo, que não são configurados corretamente e, em muitos casos, causam tempos de espera desnecessários tanto para os veículos, como para pedestres. Isso ocorre porque o sistema de controle de tráfego não é linear, utilizando métodos não determinísticos e difusos, com base em modelagem e controle que nem sempre funcionam bem.

1.2 OBJETIVOS

1.2.1 Geral

Investigar os principais algoritmos de subtração de fundo, ou do inglês *background subtractor* (BGS) e aplicá-los em um modelo computacional para o gerenciamento de tráfego em uma cidade.

1.2.2 Específicos

- i) Comparar os algoritmos utilizando as mesmas cenas gravadas em vídeo, garantindo as mesmas condições para todos os algoritmos;
- ii) Avaliar a acurácia dos algoritmos de subtração de fundo, simulando uma aplicação prática.

1.3 JUSTIFICATIVA PARA O ESTUDO DO TEMA

Atualmente, as abordagens e o monitoramento de tráfego não são eficazes o suficiente na maioria dos casos.

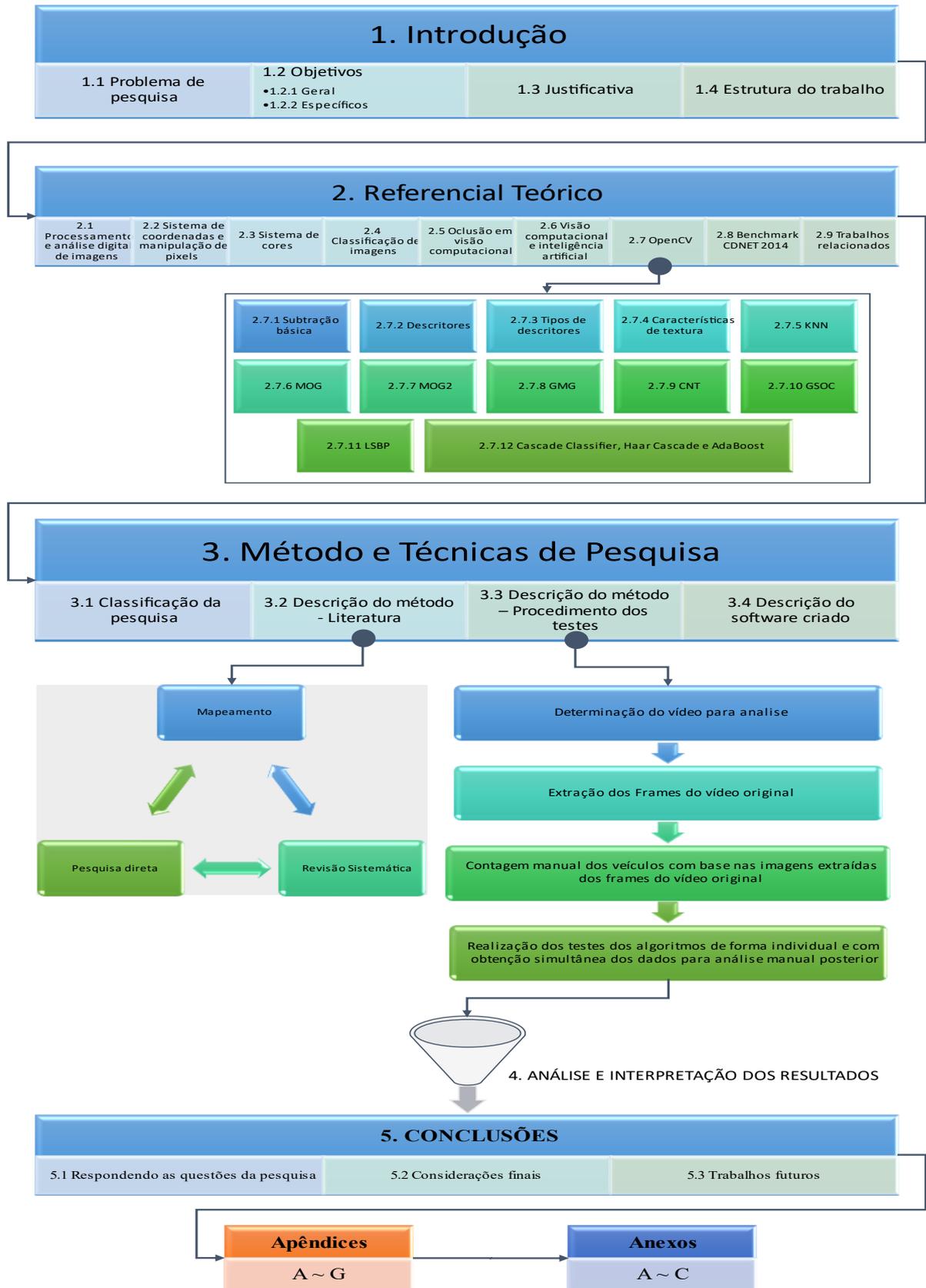
Para diminuir a dependência de autoridades de trânsito e melhorar a eficiência da fiscalização e monitoramento, pode-se usar câmeras que fornecem maior alcance e a automação de tarefas como fiscalização e controle de fluxo, além do registro de ocorrências. Portanto, aqui o sistema de visão computacional e tecnologias de *Machine Learning* (ML) podem auxiliar através do sistema de monitoramento de vídeo.

Muitos sistemas atuais são baseados em sensores indutivos, que são usados para detectar a densidade de veículos na pista, e podem se deteriorar mais rapidamente por serem instalados diretamente nos pavimentos. Estes podem ser substituídos por um sistema baseado em visão computacional, que considera como entrada, as imagens das câmeras instaladas nos semáforos de trânsito.

1.4 ESTRUTURA DO TRABALHO

Assim exposto, os procedimentos de análise e interpretação dos dados, assim como os instrumentos adotados neste trabalho e para representar da melhor forma possível este trabalho, as etapas são apresentadas por meio do Fluxograma 1.

Fluxograma 1 - Estrutura da dissertação



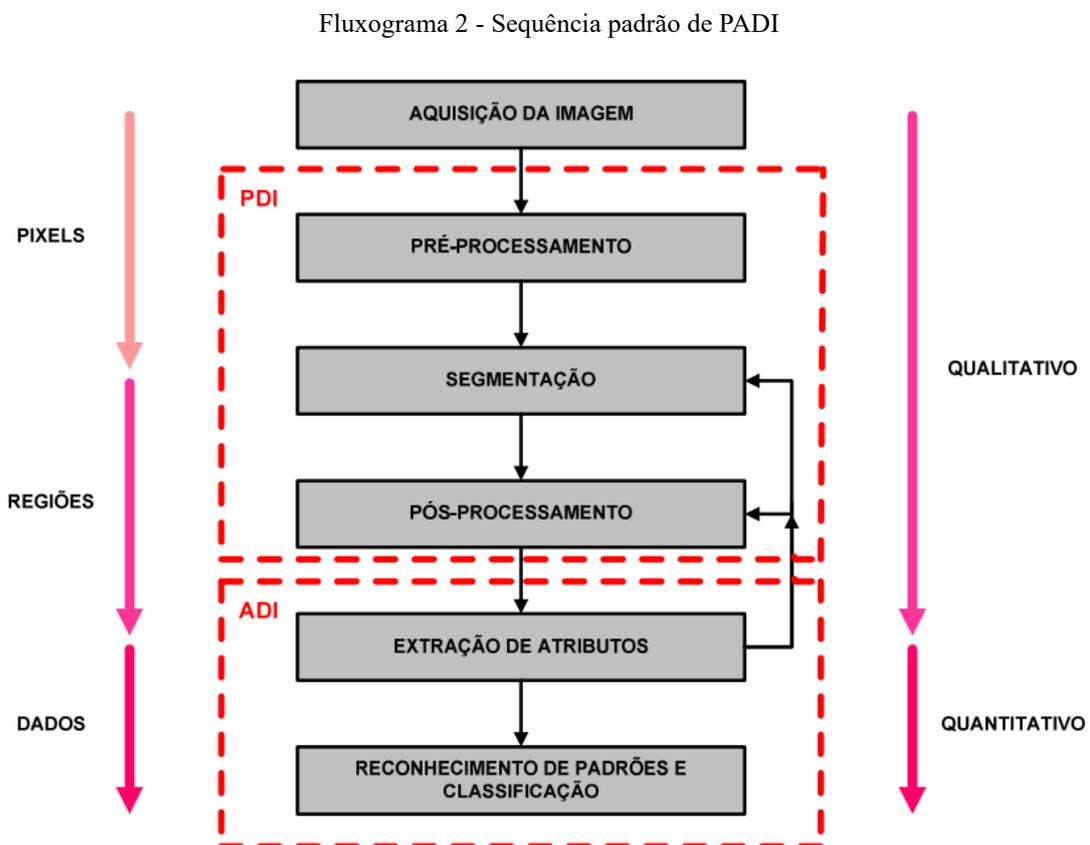
Fonte: O Autor

2 REFERENCIAL TEÓRICO

Para o entendimento básico sobre visão computacional, precisa-se entender o básico sobre imagem digital e como é formada as imagens, porém, o aprofundamento sobre os temas abordados foge ao escopo desta obra, que irá abordar a utilização de algoritmos de subtração de fundo (BGS) que compõem o OpenCV, que se é uma biblioteca multiplataforma de uso livre, tanto comercial como acadêmico.

2.1 Processamento e análise digital de imagens

No Fluxograma 2 demonstra o fluxograma apresentado por (AUGUSTO, 2012) e é dividido em três etapas: aquisição da imagem, PDI e ADI.



Segundo (AUGUSTO, 2012) o processamento de imagens é como o conjunto de duas técnicas:

- **Processamento Digital de Imagens (PDI)**, que consiste no preparo da imagem para posteriores análises, em que são realizadas operações matemáticas que alteram os valores dos pixels;

- **Análise Digital de Imagens (ADI)**, que compreende a análise quantitativa do processo a partir da qual as regiões, partículas e objetos identificados na imagem são medidos.

O autor anterior adota além destes termos, o **Processamento e Análise Digital de Imagens (PADI)** para quando estiver empregando as duas categorias.

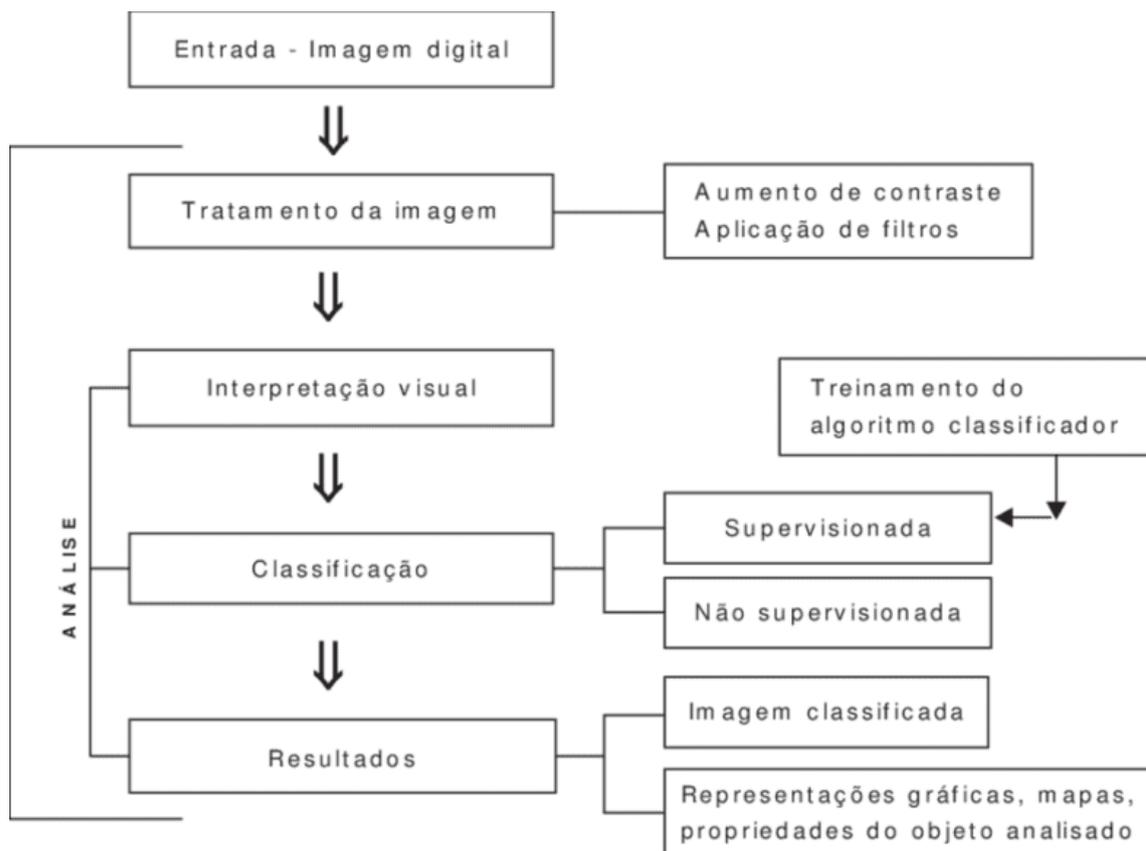
Ainda segundo (AUGUSTO, 2012) os cinco passos são:

- 1) Aquisição da imagem, com a formação e a digitalização da imagem.
- 2) Pré-processamento da imagem da etapa anterior, que serve para melhorar a imagem, para que as próximas etapas sejam bem-sucedidas.
- 3) Segmentação que é separar a imagem em partes ou objetos de interesse; que em sua maior parte o resultado da segmentação não é adequado e para corrigir esses defeitos, realiza-se o estágio de pós-processamento.
- 4) No pós-processamento é gerada a imagem para fornecer dados numéricos através da etapa de extração de atributos, que é a parte quantitativa da sequência. Nesta etapa, algumas características dos objetos nas imagens são extraídas, resultando em algum tipo de discriminação entre as classes de objetos.
- 5) O reconhecimento atribui uma descrição a um objeto, baseado na informação fornecida pelo seu descritor e em conjunto a classificação confere um significado para um conjunto de objetos reconhecidos.

No pré-processamento e na segmentação, as operações são feitas nos pixels das imagens e que resulta em imagens com objetos ou regiões de pixels contíguos de valores iguais. Essas medidas geram dados que são feitos na extração de atributos e estas medidas são utilizadas no fim do processo, na etapa de reconhecimento de padrões e classificação, de forma a obter dados mais complexos. As etapas de segmentação, pré-processamento e pós-processamento fazem parte do PDI, enquanto as etapas de extração de atributos e reconhecimento de padrões e classificação fazem parte do ADI.

Ainda para (AUGUSTO, 2012) que ressalta que a sequência é adaptada para o problema a ser resolvido e não há a obrigação de todas as etapas serem realizadas e, além disso, o fluxo poderá retroceder a qualquer outra etapa, caso o resultado obtido não seja satisfatório.

Fluxograma 3 - Processamento digital de imagem



Fonte: (GIOVANINI, 2022)

Segundo (GIOVANINI, 2022) o processamento digital de imagem é dividido conforme o Fluxograma 3 onde o pré-processamento é responsável pelo processamento inicial dos dados brutos e sub dividida em calibração, correção e filtragem; em realce a qualidade da imagem é aperfeiçoada, objetivando aumentar o contraste utilizado como pré-processamento em sistemas de reconhecimento de padrões. O autor lista os principais métodos de realce de contraste: Realce linear; mínimo/máximo; raiz quadrado; equalização de histograma, fatiamento, edição e realce logarítmico.

Já para (ANTONELLO, 2017) define como sendo uma matriz de 2 dimensões, onde cada célula dessa matriz é um pixel e exemplifica no caso de imagens em preto e branco, que possuem um valor de 0 a 255, onde: 0 para preto e 255 para branco; conforme demonstrado na Figura 1 e assim cada célula contém um inteiro de 8 bits (sem sinal) que em *Python* é definido por “uint8” que é um *unsigned integer* de 8 bits.

Figura 1 - Imagem preto e branco representada em uma matriz de inteiros

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	0	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255
2	0	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255
3	0	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255
4	0	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255
5	0	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255
6	0	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255
7	0	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255
8	0	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255
9	0	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255
10	0	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255

Fonte: (ANTONELLO, 2017)

[...] para imagens coloridas há três matrizes de duas dimensões e cada uma representa uma cor do sistema RGB; assim cada pixel é formado de uma tupla de 3 inteiros de 8 bits sem sinal no sistema (R, G, B); sendo que (0,0,0) representa o preto, (255,255,255) o branco. As imagens coloridas, são compostas normalmente de 3 matrizes de inteiros sem sinal de 8 bits, a junção das 3 matrizes produz a imagem colorida com capacidade de reprodução de 16,7 milhões de cores, sendo que os 8 bits têm capacidade para 256 valores e elevando a 3 temos $256^3 = 16,7$ milhões. (ANTONELLO, 2017, p. 8)

2.2 Sistema de coordenadas e manipulação de pixels

Para mudar a cor individualmente de cada pixel, segundo (ANTONELLO, 2017), e assim manipular cada pixel que forma uma imagem digital é importante entender o sistema de coordenadas onde o pixel mais à esquerda e acima da imagem está na posição [0,0], ou seja, na linha zero e coluna zero. O autor exemplifica através da Figura 2 o sistema de coordenadas com base em uma imagem de 300x200 pixels, ou seja, 300 colunas (largura) por 200 linhas (altura) e nesse exemplo, o pixel mais à direita e abaixo da imagem ocupa a coordenada: [199,299].

Figura 2 - Sistema de coordenadas de imagens - Linha e Coluna

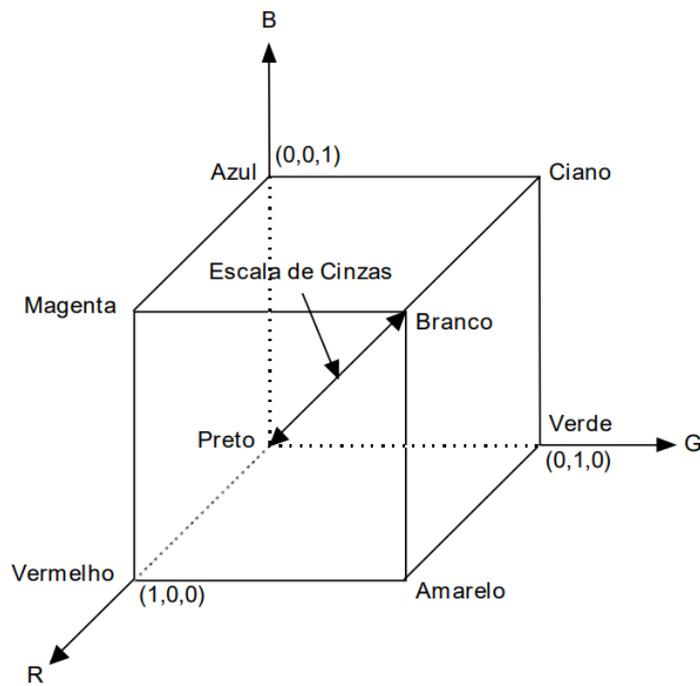
	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9
L0	0	0	0	0	0	0	0	0	0	0
L1	0	50	50	50	50	50	50	50	50	0
L2	0	50	100	100	100	100	100	100	50	0
L3	0	50	100	150	150	150	150	100	50	0
L4	0	50	100	150	200	200	150	100	50	0
L5	0	50	100	150	200	200	150	100	50	0
L6	0	50	100	150	150	150	150	100	50	0
L7	0	50	100	100	100	100	100	100	50	0
L8	0	50	50	50	50	50	50	50	50	0
L9	0	0	0	0	0	0	0	0	0	0

Fonte: (ANTONELLO, 2017, p. 9)

2.3 Sistemas de cores

Segundo (FILHO e NETO, 1999), que apresentam os modelos mais utilizados para a representação de cores, entre eles o RGB (*Red, Green, Blue*) que é baseado em coordenadas cartesianas e que como demonstra a Figura 3, representa três de seus vértices como as cores primárias, três outros com as cores secundárias, o vértice da origem é o preto e o mais afastado da origem é a cor branca. Os autores afirmam que o modelo RGB é o mais utilizado em câmeras e monitores de vídeo.

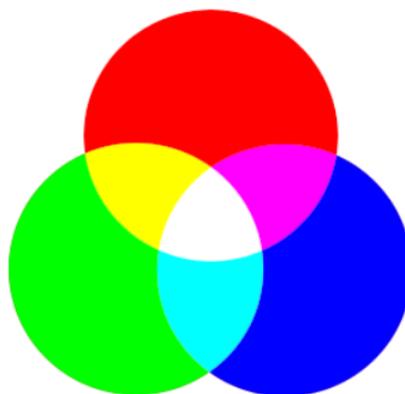
Figura 3 - Modelo RGB



Fonte: (FILHO e NETO, 1999, p. 121)

Segundo (ANTONELLO, 2017), que afirma que o OpenCV trabalha com o sistema de cores tradicionalmente conhecido como RGB (Red, Green, Blue), porém com a ordem inversa, ou seja, BGR (Blue, Green, Red) compondo uma tupla de cores de pixels. O autor ainda demonstra outros espaços de cores, como visto na Figura 4 e Figura 5.

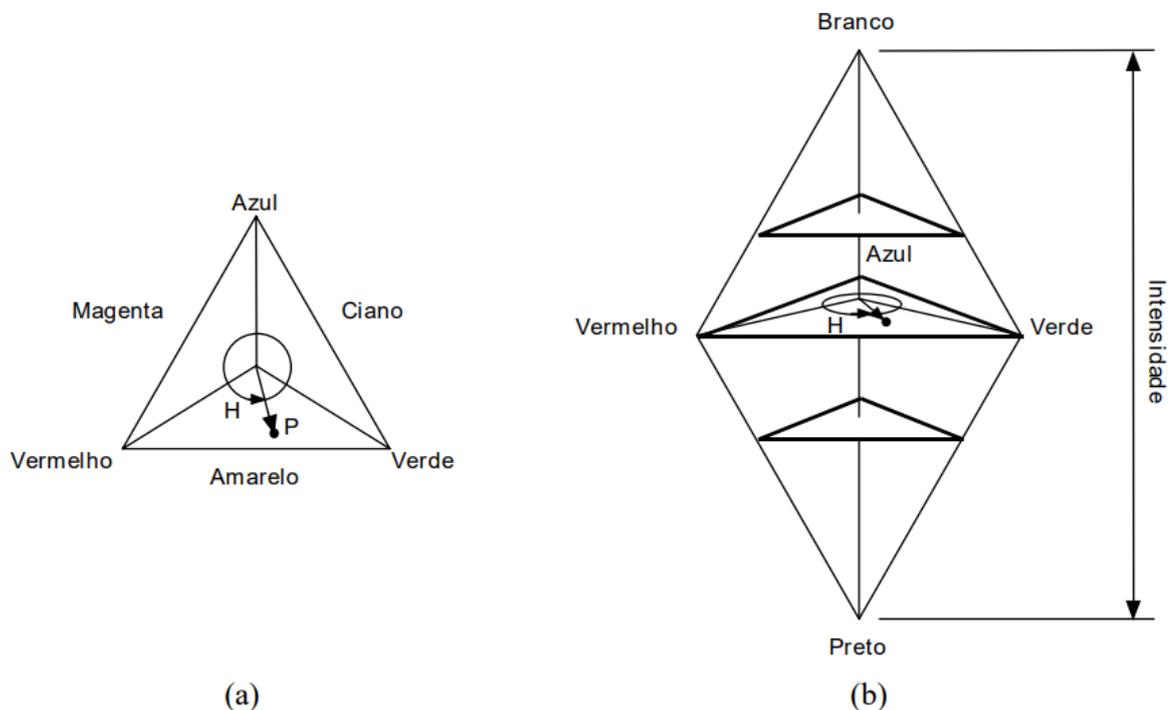
Figura 4 - Sistema RGB de cores



Fonte: (ANTONELLO, 2017, p. 23)

Segundo (FILHO e NETO, 1999), que apresentam o modelo HSI (*Hue, Saturation, Intensity*) que ocasionalmente pode ser denominado como HSV (*Hue, Saturation, Value*), que segundo os autores é de grande interesse, pois permite separar os componentes de matiz, saturação e intensidade dos dados de cor em uma imagem. Segundo os autores, a utilização deste sistema é mais comum em sistemas de visão artificial baseados em modelos de percepção de cor, como o exemplo dado por eles: sistema automatizado de colheita de frutas, onde se precisa determinar se as frutas estão maduras para serem colhidas com base na cor de sua casca. Sua representação geométrica pode ser vista na Figura 5 item B e cujos cortes horizontais produzem os triângulos do item A desta mesma figura, onde os vértices contêm as cores primárias e o centro é a combinação destas cores em proporções idênticas e estarão mais próximas do preto ou do branco, conforme a altura do corte efetuado. Na Imagem 2 encontra-se uma comparação entre os modelos de cores.

Figura 5 - Modelo HSI



Fonte: (FILHO e NETO, 1999, p. 123)

Imagem 2 - Espaços de cores: Cores, Tons de Cinza, L*a*b e HSV



Fonte: (ANTONELLO, 2017, p. 23)

2.4 Classificação de imagens

Segundo (GIOVANINI, 2022), a classificação de imagens é feita a amostragem e atribuindo-se classes as diferentes feições na imagem, podem segundo o autor, ser divididos em classificadores “pixel a pixel” e por regiões. Sendo que os primeiros utilizam o espectro de cada pixel para encontrar regiões homogêneas e podem ser separados em dois métodos: estatísticos (regras da teoria de probabilidade) e os determinísticos (não utilizam probabilidade). Os classificadores por regiões procuram além de espectros de cada pixel, procuram reconhecer áreas homogêneas das imagens e se baseiam nas propriedades espectrais e espaciais destas.

2.5 Oclusão em visão computacional

Oclusão na visão computacional, segundo (CHILUKURI, YI e SEONG, 2022), é a obstrução de um objeto devido a um outro objeto ou obstáculo intermediário em sobreposição

ao mesmo e que podem ser causadas por uma variedade de fatores, como a configuração do sensor, condições ambientais ou eventos diversos; completa que na visão computacional, as oclusões podem dificultar a detecção e o rastreamento preciso de objetos. Para resolver esse problema, os pesquisadores desenvolveram técnicas para lidar com oclusões em sistemas de detecção de objetos, bem como técnicas de aumento de dados baseadas em oclusão para treinar modelos de aprendizado profundo.

Além disso, algoritmos de rastreamento de objetos em tempo real foram desenvolvidos usando o OpenCV para ajudar a remover oclusões; conforme (STACKOVERFLOW, 2018), atesta que a oclusão ocorre se um objeto que está sendo rastreado estiver oculto por outro objeto; como demonstra a Figura 6; com as duas pessoas passando uma pela outra, o livro recobrendo o rosto e assim, o problema, neste caso, é o que fazer quando um objeto desaparece e reaparece novamente. Se for utilizada uma camera de alcance, então oclusão são áreas em que não tem nenhuma informação e ainda, o mesmo pode ocorrer em uma imagem estéreo se houver partes da cena que são vistas apenas por uma das duas câmeras e nenhum dado de intervalo pode ser coletado a partir desses pontos.

Figura 6 - Exemplos de oclusão



(a) Occluded Face 1



(c) Person

Fonte: (STACKOVERFLOW, 2018)

Segundo (PINHO, TAVARES e CORREIA, 2005), afirmam que não é permitida a alteração do número de pontos a serem seguidos ao longo da sequência de imagens, mas casos de oclusão de pontos são frequentes na análise e seguimento de movimento, pelo que tal situação deverá ser ponderada de forma adequada. Conforme (THAKOR e ASKHEDKAR, 2014) afirmam que além disso, algoritmos de rastreamento de objetos em tempo real foram desenvolvidos usando o OpenCV para ajudar na remoção de oclusões.

Conforme (SOLAWETZ, 2020) apresenta uma visão geral das técnicas de oclusão e discute como a ela é baseada em técnicas de aumento de dados e conclui que tais técnicas podem ser usadas para melhorar a resiliência do modelo de visão computacional, ensinando-o a generalizar os recursos canônicos da imagem analisada; afirmando ainda, que a pesquisa na área continua a ultrapassar os limites de estratégias mais eficazes e que a oclusão é uma estratégia de aumento de dados importante, por explorar a possibilidade do modelo se adaptar a determinados recursos do conjunto de treinamento.

2.6 Visão computacional e inteligência artificial

Em 1956 é através da “*Dartmouth Summer Research Project on Artificial Intelligence*”, segundo (OPENCADD, 2021); (C.CHRISTOFHER, 2021) e (PÚBLICO, 2021), que afirmam que datam da década de cinquenta o início das experiências em visão computacional, é quando foram usadas algumas das primeiras redes neurais para detecção de um objeto e da sua classificação, em categorias bem simples, como quadrados e círculos. Aproximadamente vinte anos depois, ocorre o seu primeiro uso comercial, onde a visão computacional foi responsável por interpretar textos manuscritos e digitados; e assim, através do reconhecimento ótico dos caracteres, contribuindo no auxílio aos deficientes visuais.

Entre os pioneiros, destaca-se o trabalho de (MARR, 1976), que a partir da neurofisiologia da visão humana, desenvolveu o modelo de interpretação que passou a ser usado na visão computacional, estabelecendo os níveis de compreensão necessários à computação da visão estereoscópica; por meio da formulação de um algoritmo que aplica um programa de interpretação visual. Suas pesquisas foram o ponto de partida para outros estudos em visão computacional e permitiu que fosse estabelecido o primeiro conceito para a área.

Conforme (BESL e JAIN, 1985) afirmam que o objetivo principal da pesquisa em visão computacional é demonstrar uma capacidade visual semelhante ao ser humano, fazendo a máquina “sentir” o ambiente em seu campo de visão, compreender o que está sendo sentido e realizar ações apropriadas através de programas computacionais.

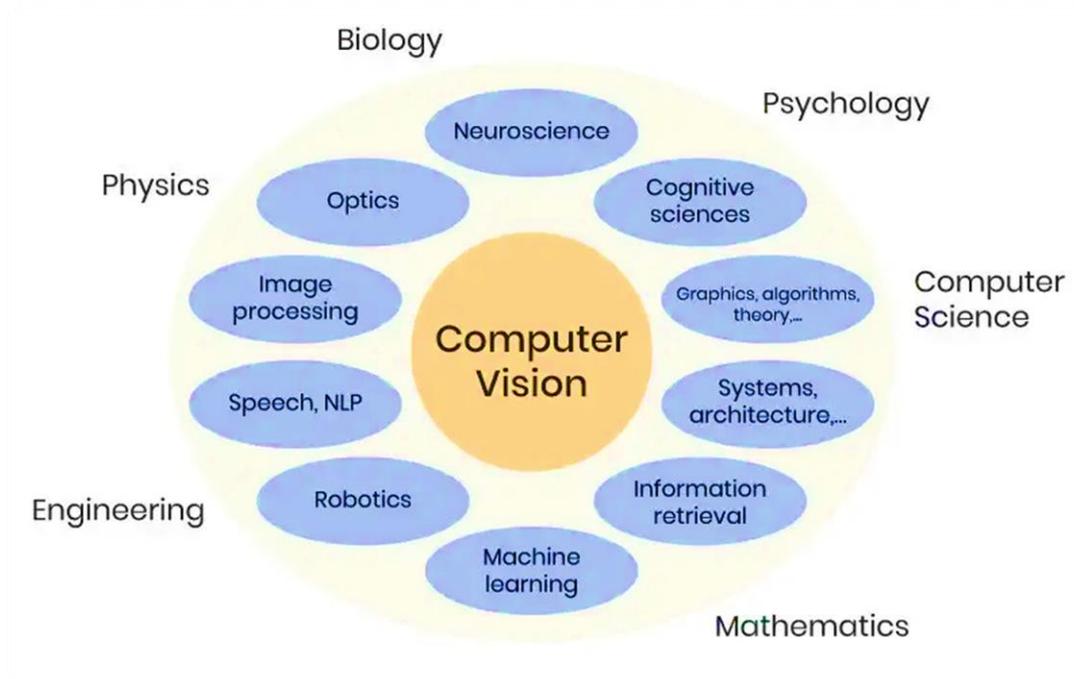
Para (FU, GONZALEZ e LEE, 1987), a visão computacional é um dos ramos da inteligência artificial que estuda o processamento de imagens do mundo real por um computador. Investigando recursos para capacitar às máquinas a capacidade de interpretar visualmente informações, ou seja, enxergar.

Segundo (BLACKMORE, 1981) descreve como a atividade visual ocorre pelo funcionamento das células visuais, sendo que a arquitetura destas é compatível com a de um computador; essa arquitetura permite um acesso às informações de uma forma sequencial e paralela; assim sendo, a forma sequencial é caracterizada pela transferência de dados de tempos em tempos; por exemplo: dois sinais são apresentados seguidamente, sendo que o segundo sinal será processado somente se o processamento do primeiro sinal já estiver completado, e a paralela é dada simultaneamente, ou seja, o processamento é o sincronizado.

Para (WANGENHEIM, 2005) a visão por computador é responsável pelo desenvolvimento dos métodos e teorias, que são destinados de forma automática extrair informações a partir de imagens. Imitando assim o comportamento humano de reconhecer as imagens e tudo o que as envolve, além de tomar decisões com base nisso; assim trabalhando com modelagem e replicação da visão biológica por meio de hardwares e softwares.

Para (BERGMARK, 2018), a visão computacional é um campo da ciência da computação, que permite que os computadores vejam, identificando e processando imagens como a visão humana, em seguida, fornecendo dados para tomada de ações; é como transferir inteligência e instintos humanos a um computador. Desta forma ela está ligada à inteligência artificial, pois o computador interpreta o que vê e, em seguida, executa diversas análises mais apropriadas ou toma uma ação conforme sua programação. Na Figura 7 exemplifica o entendimento do autor sobre as áreas que abrangem a visão computacional e na Figura 8 os três níveis de processamento de imagens.

Figura 7 - Áreas que formam a Visão Computacional



Fonte: (BERGMARK, 2018)

Figura 8 - Níveis de processamento de imagens



Fonte: Adaptado de (BERGMARK, 2018)

Segundo (TECMUNDO, 2021) e (EMBRATEL, 2021), foi na década de noventa, com o surgimento, avanço e amadurecimento da internet, aliado com a presença de muitas imagens disponibilizadas na internet foi possível que programas de reconhecimento facial e um conjunto de dados, cada vez mais robusto, se disseminassem e as máquinas começaram a terem mais facilidade em reconhecer pessoas em vídeos e fotos.

Para a fabricante NVIDIA (NVIDIA, 2022), a visão computacional é considerada um subgrupo da IA e do DL, além de treinar as CNNs para desenvolverem algo similar a visão biológica em soluções computacionais; podendo ainda, incluir treinamentos específicos de CNNs para a segmentação, classificação e detecção; usando para isso, imagens e vídeos como a fonte de dados, além das necessidades das aplicações; conforme ilustra o Quadro 1.

Quadro 1 - Comparativo: Segmentação x Classificação x Detecção

Segmentação	Classificação	Detecção
É a classificação de pixels em determinadas categorias, exemplo: "carro", "estrada" ou "pedestres". Muito usada em aplicações de veículos autônomos para indicar estradas, carros e pessoas, sendo uma técnica de visualização que facilita o entendimento sobre o que os computadores fazem.	Usada para determinar o que uma imagem contém. Redes neurais podem ser treinadas para identificar com muita precisão, muitos elementos, como exemplo: cães ou gatos.	Permite aos computadores localizarem onde há objetos. Em muitas aplicações, a CNN posiciona caixas retangulares delimitadoras ao redor da região de interesse (ROI), contendo o objeto. Um detector também pode ser treinado para encontrar carros ou pessoas em uma imagem.
Delimita objetos	É um gato ou um cachorro?	Onde o objeto está no ambiente?
Usada em veículos autônomos	Classifica com precisão	Reconhece objetos para garantir a segurança

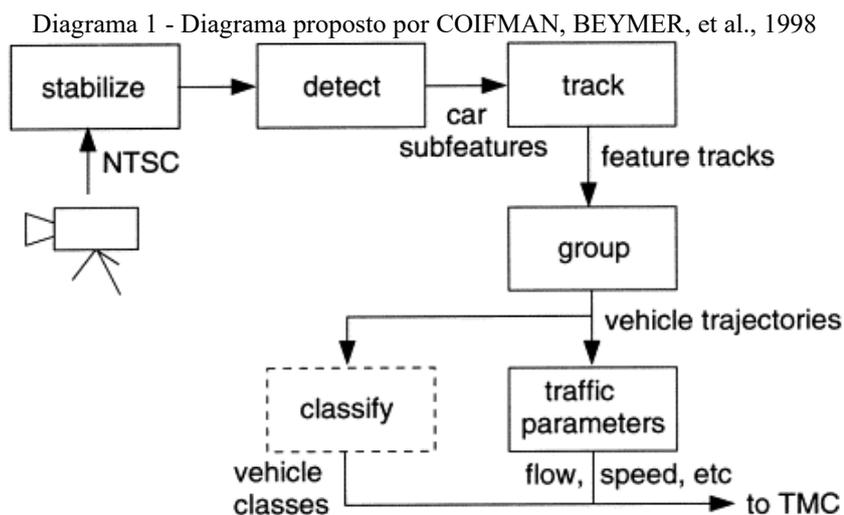
Fonte: Adaptado de (NVIDIA, 2022)

(JAHNE e HAUBECKER, 2000) defendem que em um sistema de visão, os componentes básicos são: a **fonte de radiação**, pois sem a radiação nada será observado; a **camera**, que capta a radiação emitida pelos objetos e que as origens dessas radiações possam ser identificadas e processadas; o **sensor**, que converte a densidade radioativa recebida em um sinal adequado para tratamento; uma **unidade de processamento** associado à memória do sistema; e por fim; de **agentes** que reagem ao resultado da observação final. Além de afirmarem que a meta do sistema de visão computacional é converter as imagens captadas em sinais digitais, e dessa forma, possamos obter informações sobre onde ele está (*geometria e posicionamento com base em eixos cartesianos*), aquilo que é (*ser ou que objeto é*), ou quais suas propriedades. Simplificando o sistema de visão computacional; requer a entrada de dados (imagem), seja esta obtida por meio sensores, camera ou vídeos.

O emprego de sistema de controle baseado em imagens digitais é teorizado também por (GONZALEZ e WOODS, 2007) que descrevem os conceitos básicos e metodologias para processamento, desenvolvendo os fundamentos para realce e restauração, passando pela codificação das imagens, segmentações, descrições, além do reconhecimento e interpretação.

Em 1993, (KOLLER, DANIILIDIS e NAGEL, 1993) propuseram um sistema de rastreamento baseado em modelo que usa parâmetros de geometria do veículo como técnica de comparação. O sistema identifica o provável veículo em movimento, extraíndo e identificando os parâmetros na imagem.

Para (COIFMAN *et al.*, 1998) que introduziram uma técnica de rastreamento com base em região, que identifica as regiões de interesse na imagem real, com cada amostra de veículo e os rastreia. Principalmente a técnica de subtração de fundo foi usado no sistema. No entanto, esta técnica falha em situações de tráfego muito intenso, devido à sobreposição de veículos na imagem capturada. Nesta condição, o sistema detecta os veículos como um único BLOB na imagem de primeiro plano, o que reduz a precisão. A técnica baseada em filtro de Kalman ajuda a visão de fundo a desenvolver a visão em relação a iluminação, o clima ou as mudanças de horário do dia. O rastreamento de contorno ativo é outra abordagem que usa modelos de contorno para realizar o rastreamento. Nesta abordagem, os autores conseguiram reduzir a complexidade de cálculos. Conforme demonstrado no Diagrama 1.



Fonte: (COIFMAN *et al.*, 1998)

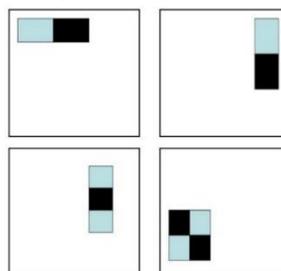
Para (NEVES e PELAES, 2001) normalmente as técnicas baseadas em regiões tratam da extração das informações de cor e textura dos pixels na imagem capturada, onde com base na cor dos pixels capturados deve-se segmentar a imagem em regiões consideradas ou não de interesse, para isso, o primeiro passo é a sua segmentação; onde o PDI, divide a imagem de entrada em partes e essa quantidade de divisões efetuadas depende do contexto do problema; geralmente isso é uma das tarefas mais difíceis em PDI, porém simplifica a análise computacional da imagem analisada. Essa segmentação pode seguir duas propriedades: descontinuidade e similaridade.

Ainda segundo os autores anteriores, que descrevem no caso de descontinuidade; a segmentação da imagem é feita baseando-se nas alterações bruscas de intensidade, como a detecção dos contornos, na detecção de pontos isolados e na detecção de linhas. E para a similaridade; essa segmentação é realizada com base na similaridade entre os pixels, seguindo o critério definido como: binarização, crescimento de regiões (*Region Growing*), divisão e junção das regiões (*Split & Merge*), aglomeração (*Clustering*) e limiares (*Thresholding*)

A próxima abordagem, extrai sub recursos e os rastreia na imagem correspondente, que pode ser de importância analítica, para (FATHY e SIYAL, 1995) esta abordagem oferece uma vantagem de encontrar alguns recursos diferenciáveis na imagem pelo qual o sistema pode detectar o veículo com mais eficiência, mesmo quando ocorre uma sobreposição.

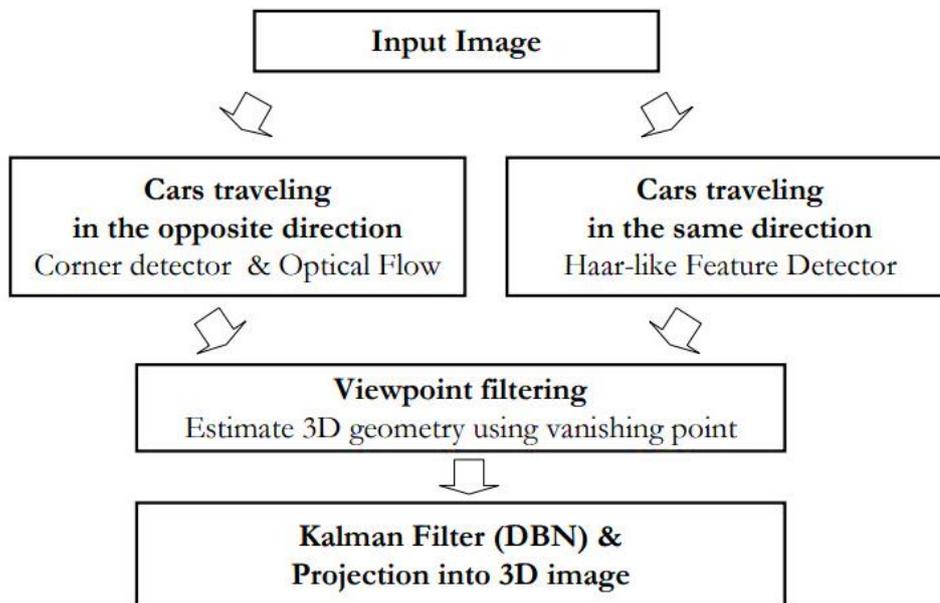
(CHOI, 2006) afirma que para um veículo autônomo funcione baseado em visão computacional, este requer algoritmos de detecção de veículos e de outros objetos que possa encontrar na estrada; o processamento da sequência de imagens capturadas pelos seus sensores, precisam encontrar veículos em tempo real. Utilizando-se de detecção de características do tipo Haar foi possível detectar veículos com alta precisão e em tempo real por 11 quadros por segundo. No Diagrama 2 e Figura 9 demonstram o trabalho.

Figura 9 - Filtro Haar



Fonte: (CHOI, 2006)

Diagrama 2 - Algoritmos de detecção de veículos na estrada



Fonte: (CHOI, 2006)

Ainda segundo (CHOI, 2006) um recurso do tipo Haar é definido por um dos retângulos acima (ou variações) e um valor limite. É dada uma imagem “I”, calcula-se a soma das intensidades da região escura e da região clara. Se a soma da intensidade na região escura for maior (ou menor) que a soma da intensidade na região clara pelo limiar, a região é determinada como exemplo positivo. Caso contrário, a região é determinada como exemplo negativo.

Segundo o autor anterior, que empregando o *AdaBoost*, que é um algoritmo de aprendizado de máquina inventado por (FREUND e SCHAPIRE, 1985) que por ser um algoritmo meta-heurístico é utilizado para aumentar a performance de outros algoritmos de aprendizagem; conseguiu detectar os veículos na mesma direção, devido ao *AdaBoost* ter uma abordagem de detecção rápida e precisa; além de ser um recurso do tipo Haar, é um classificador, porque pode classificar todo o grupo de imagens entre: imagens positivas e negativas. Usando como um classificador do algoritmo *AdaBoost*, assim como (VIOLA, JONES e SNOW, 2003) o fizeram para detecção de face, mesmo que um recurso satisfaça um critério simples (a taxa de erro é menor (0,5)).

(CHAWLA *et al.*, 2002) e (LIU *et al.*, 2016) usaram um detector de recurso semelhante ao Haar para detecção dos veículos, que possuem uma boa precisão. Funciona em intensidades de pixel na janela de detecção, levando a área retangular adjacente em um local específico e soma a intensidade de pixels em cada área. Então, as diferenças dessas somas são calculadas o que ajuda a classificar subseções dessa imagem.

Objetos em movimento, como veículos e pedestres são considerados alvos (o que se deseja rastrear). Estes objetos encontram-se na maioria das vezes em movimento inseridos em um ambiente observado através de câmeras; segundo os autores (DEDEOGLU, 2004); (MCIVOR, 2000); (PICCARDI, 2004); (CHEUNG e KAMATH, 2004) e (HERRERO e BESCÓS, 2009) a subtração de fundo é uma técnica usada para segmentar o movimento em cenas estáticas e tenta detectar desta forma, regiões em movimento, subtraindo uma imagem atual (pixel a pixel) de uma imagem de fundo (referência), criada pela média das imagens ao longo de um tempo em um período de inicialização. Os pixels onde a diferença está acima de um limite são classificados como primeiro plano. Depois de criar um mapa de pixels de primeiro plano, algumas operações de pós-processamento morfológico, como erosão, dilatação e fechamento, são realizadas para reduzir os efeitos do ruído e aprimorar as regiões detectadas. O plano de fundo de referência é atualizado com novas imagens ao longo do tempo para se adaptar às mudanças dinâmicas de cena.

A segmentação do movimento segundo (SIEBEL e MAYBANK, 2004) consiste na separação dos objetos que se movimentam da parte estática ou fundo da cena, isso ocorre devido a uma sequência de Frames de entrada, onde as imagens são detectadas pela detecção de movimento; já na saída o resultado é as imagens binarizadas, onde destacam-se as regiões *BLOBS* em movimento das regiões estáticas.

Ainda segundo os autores, para realizar a segmentação de movimento a partir de um *Frame* de entrada, o detector de movimento é mantido como um modelo de fundo da cena (*background model*), isto é, uma máscara/molde do que aparenta ser o ambiente sem a presença de pessoas ou objetos em movimento. Nesse modelo, é aplicado um algoritmo para classificar cada pixel do *ROI* como um pixel que pertence a um objeto alvo (algo em movimento). Os modelos utilizados sofrem mudanças que podem afetar o desempenho, essas podem ser de origem climáticas, iluminação, sombras, superfícies reflexivas, objetos com movimentos oscilatórios com baixa amplitude, como galhos e folhas de vegetação sob ação do vento, são alguns dos exemplos, dessa maneira fazem da segmentação de movimento uma tarefa difícil,

podendo comprometer a confiabilidade e o desempenho do algoritmo utilizado para segmentação.

Segundo (LARA, 2007), não há algoritmo que seja robusto para se adaptar as diversificadas situações ambientais, porém, quanto menos suposições em relação ao ambiente não controlado, esse algoritmo necessita ser muito mais robusto e dessa forma, devemos ter consciência do tipo de aplicação que iremos utilizar para executar a segmentação.

O rastreamento é baseado em algoritmo para essa finalidade, na literatura pode-se encontrar diversos critérios para classificação destes algoritmos. Em especial com base no movimento de pessoas, assim como os autores (MOESLUND, HILTON e KRUGER, 2006); (WANG e BRANDSTEIN, 1998) e (ELGAMMAL, HARWOOD e DAVIS, 2000) detalham sobre os tipos de algoritmos de rastreamento.

2.7 OpenCV

Em resumo o *Open Source Computer Vision Library* (OpenCV) é uma biblioteca gratuita de software de visão computacional e aprendizado de máquina multiplataforma e foi construído para fornecer uma infraestrutura comum para aplicações em visão computacional e para acelerar o uso da percepção de máquina em produtos comerciais, composto por mais de dois mil e quinhentos algoritmos otimizados, incluindo um conjunto abrangente de visão computacional clássica e de última geração e de aprendizado de máquina. Ainda segundo a (OPENCV.ORG (A), 2022); sendo ainda licenciado sob o formato Apache 2; o que o torna mais fácil para as empresas utilizarem e modificarem o código; que inclui além dos algoritmos de visão computacional, uma estrutura modular, o que significa que o pacote inclui várias bibliotecas compartilhadas ou estáticas conforme a necessidade de seu usuário.

Entre as diversas possibilidades de utilização, como demonstrado por (BOESCH, 2023) estão os casos de uso da visão computacional bastante populares em vigilância por vídeo e segurança, como demonstra a Imagem 3 ou em carros autônomos, o uso em fabricação industrial e análise de varejo.

Imagem 3 - Sistema de visão computacional na detecção de veículos, identificando matrículas e fornecendo informações sobre a marca, cor, tipo e posição do veículo.



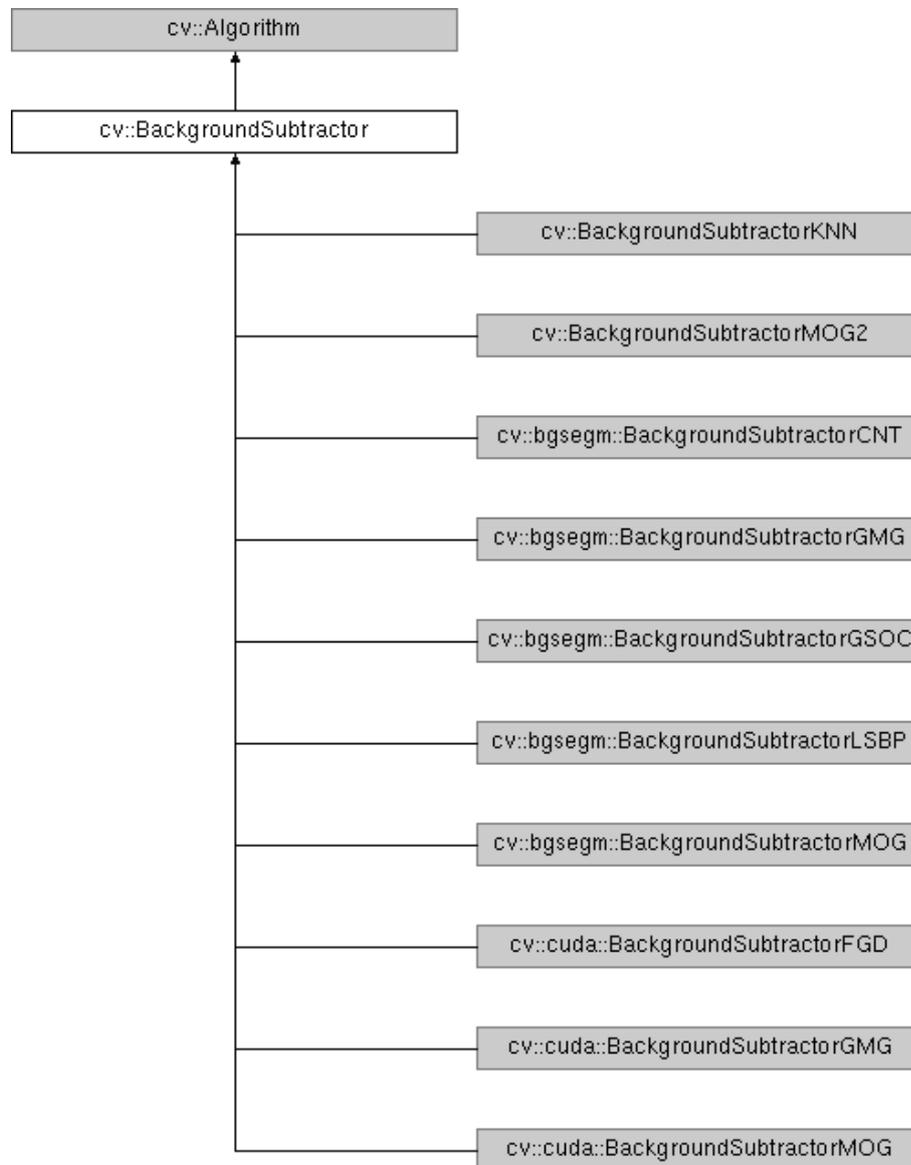
Fonte: (BOESCH, 2023)

[...] A biblioteca é amplamente utilizada em empresas como Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota que empregam a biblioteca, existem muitas startups, como Applied Minds, VideoSurf e Zeitera; grupos de pesquisa e por órgãos governamentais e possui mais de 2500 algoritmos otimizados, que incluem um conjunto abrangente de algoritmos clássicos e de última geração de visão computacional e aprendizado de máquina; que podem ser usados para detectar e reconhecer rostos, identificar objetos, classificar ações humanas em vídeos, rastrear movimentos de câmera, objetos em movimento, extrair modelos 3D de objetos, produzir nuvens de pontos 3D de câmeras estéreo, costurar imagens para produzir uma imagem de alta resolução de uma cena inteira, encontrar imagens semelhantes de um banco de dados de imagens, remover olhos vermelhos de imagens tiradas usando flash, seguir os movimentos dos olhos, reconhecer o cenário e estabelecer marcadores para sobrepô-lo com realidade aumentada, etc. O OpenCV tem mais de 47 mil pessoas da comunidade de usuários e um número estimado de downloads superior a 18 milhões. Os usos implantados do OpenCV abrangem a gama de costurar imagens de “streetview”, detectar intrusões em vídeo de vigilância em Israel, monitorar equipamentos de minas na China, ajudar robôs a navegar e pegar objetos na Willow Garage, detecção de acidentes de afogamento em piscinas na Europa, executar arte interativa na Espanha e Nova York, verificar pistas de detritos na Turquia, inspecionar rótulos em produtos em fábricas em todo o mundo até a detecção rápida de rosto no Japão. Ele tem interfaces C ++, Python, Java e MATLAB e suporta Windows, Linux, Androide Mac OS, se inclina principalmente para aplicativos de visão em tempo real e aproveita as instruções MMX e SSE quando disponíveis. Uma interface CUDA e OpenCL completa

está sendo ativamente desenvolvida. Existem mais de 500 algoritmos e cerca de 10 vezes mais funções que compõem ou suportam esses algoritmos. O OpenCV é escrito nativamente em C++ e tem uma interface modelada que funciona perfeitamente com contêineres STL.[...] (OPENCV.ORG (A), 2022)

Entre os recursos disponíveis: o módulo de análise de vídeo que inclui a estimativa de movimento, subtração de fundo e algoritmos de rastreamento de objetos. Para a subtração de fundo, ele conta com os módulos conforme demonstrado no Diagrama 3 e que segundo a documentação do OpenCV disponibilizada em (OPENCV.ORG (2), 2022), a classe é usada apenas para definir a interface comum para toda a família de algoritmos de segmentação de plano de fundo/primeiro plano no OpenCV.

Esses algoritmos podem ser usados para executar várias tarefas, como classificação de imagens e detecção de objetos, como (PACKT, 2022) descreve alguns dos atributos e algoritmos, utilizados para essas funções e dando exemplos práticos no campo da visão computacional; assim como em (PYIMAGE SEARCH, 2022) e para (GEEKFORGEEKS, 2022) ela desempenha papel importante na operação em tempo real, o que é muito importante nos sistemas atuais e no que se propõem nesse trabalho.

Diagrama 3 - Herança para *BackgroundSubtractor*

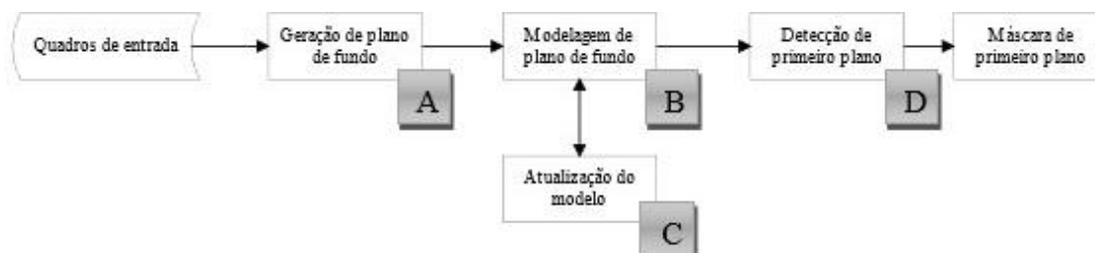
Fonte: (OPENCV.ORG (B), 2022)

2.7.1 Subtração básica

A subtração de fundo é uma técnica usada na visão computacional para detectar objetos em movimento em uma sequência de vídeo.

Segundo (MURZOVA, 2021) a subtração de fundo é um problema básico para detecção de alterações em vídeos e costuma ser a primeira etapa em aplicativos de visão computacional, conforme demonstra o Diagrama 4.

Diagrama 4 - Subtração básica do plano de fundo



Fonte: Adaptado de (MURZOVA, 2021)

Nota:

A = Há o processamento de N frames na geração da imagem de fundo;

B = Define o modelo que representa o fundo;

C = Atualiza o modelo de fundo, lidando com as mudanças que ocorrem ao longo do tempo;

D = Divide os pixels em conjuntos de fundo ou primeiro plano.

Na maioria dos casos, a tarefa de marcar entidades em primeiro plano desempenha papel importante no pré-processamento de vídeo, sendo a fase inicial dos aplicativos de visão computacional; as etapas iniciais contêm a subtração de fundo (“*BackgroundSubtractor*” ou BS), o que ajuda a obter identificações relativamente grosseiras e rápidas dos objetos no fluxo de vídeo para que seu manuseio seja ainda mais sutil para a aplicação principal. Os métodos de subtração em segundo plano resolvem a tarefa de extração em primeiro plano criando um modelo de plano de fundo. O *pipeline* completo do BS pode conter as seguintes fases:

geração de plano de fundo: processamento de N quadros para fornecer a imagem de plano de fundo;

modelagem de plano de fundo: define o modelo para representação de plano de fundo;

atualização do modelo em segundo plano: introduz o algoritmo de atualização do modelo para lidar com as alterações, que ocorrem ao longo do tempo;

detecção de primeiro plano: dividindo pixels em conjuntos de plano de fundo ou primeiro plano.

A saída de subtração de plano de fundo consiste em uma máscara binária, que separa os pixels do quadro em dois conjuntos: pixels de primeiro plano e de plano de fundo.

O GSOC e os outros métodos de subtração em segundo plano (exceto MOG2 e KNN) estão localizados nos módulos extras e exigem que a biblioteca *OPENCV-CONTRIB* seja instalada.

2.7.2 Descritores (características)

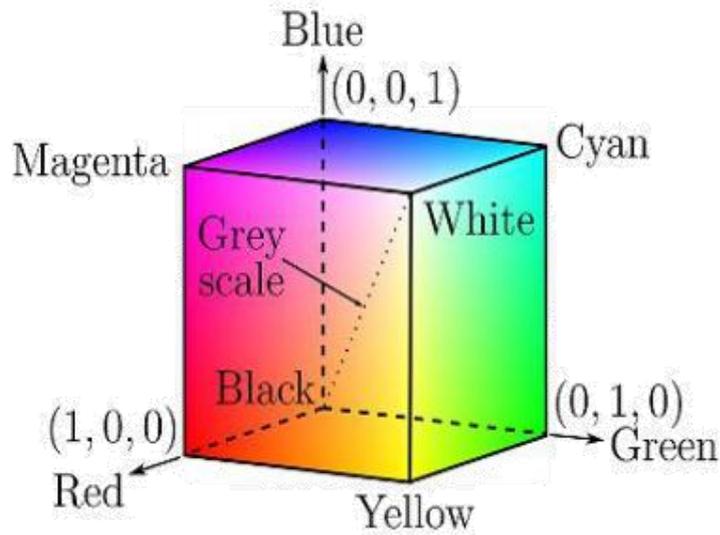
Ainda segundo (MURZOVA, 2021) os descritores definem a região da imagem capturada no quadro de vídeo atual para seu mapeamento com um modelo de plano de fundo conhecido; com objetivo de comparar e distinguir a região do fundo ou do primeiro plano. Podendo ser feito, por exemplo, com descritores de cor, textura e borda. O design do algoritmo para BS, incluindo uma combinação dos recursos, deve ser baseado na análise inicial do campo do objeto. É necessário considerar os possíveis fatores de maior desafio: iluminação, oscilações, movimento de objetos e outros.

2.7.3 Tipos de Descritores

Para (MURZOVA, 2021) os valores de pixel dos quadros estão disponíveis durante o processamento de vídeo, o cálculo dos descritores de domínio de pixel é difundido nos algoritmos de BS e os populares são:

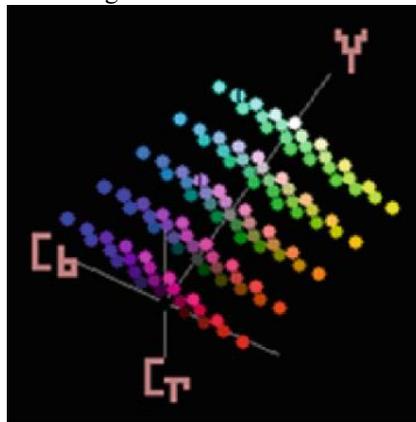
Cor: características descritivas do objeto, os componentes do espaço de cores RGB (como demonstra a Figura 10) estão firmemente conectados, reagindo às mudanças de iluminação e não há brilho e separação de cromo (como no YCrCb demonstrado na Figura 11). Os recursos de cor são sensíveis à iluminação, camuflagem e sombras, que podem afetar a aparência de objetos em movimento. Essa é a razão pela qual geralmente eles são combinados com outros recursos para se obter maior robustez.

Figura 10 - Cubo de cores RGB



Fontes: (SILVA, SISWANTO e ISTIADI, 2022, p. 82) e (STRATMANN, 2022)

Figura 11 - Cubo YCrCb



Fontes: (HERNÁNDEZ-HERNÁNDEZ *et al.*, 2017, p. 300) adaptado da Figura 2 do original

Borda: os recursos de borda são robustos às variações de luz e bons para a detecção dos objetos em movimento, além de serem sensíveis a objetos texturizados (com graus altos e baixos).

Textura: os recursos de textura fornecem informações espaciais e são robustos à iluminação e sombras.

2.7.4 Características de textura

Ainda segundo (MURZOVA, 2021) que descreve os descritores de textura e de sua evolução:

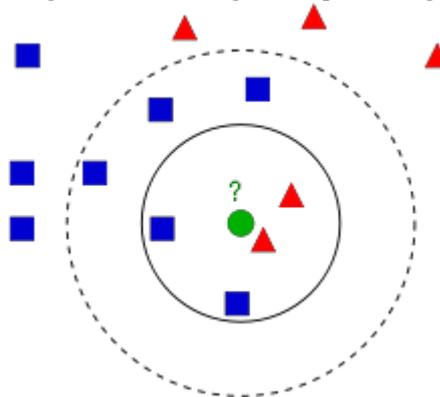
Padrões de Similaridade Binária Local (LBSP): Foi introduzido em 2013 para resolver o problema usando o limite de diferença absoluta em comparação com os pixels centrais e vizinhos, porém não é espaço-temporal e os recursos e atualizações de intensidade não são simultâneas.

Subtração em segundo plano usando LSBP (Local SVD Binary Pattern): O descritor é considerado robusto às variações de iluminação, sombras e ruídos.

2.7.5 Algoritmo: K-Nearest Neighbors (kNN)

Para (OPENCV.ORG (4), 2022) o *k-Nearest Neighbor* (kNN) é um dos algoritmos de classificação mais simples para aprendizado supervisionado, na Figura 12 é dado um exemplo do comportamento deste algoritmo, que é detalhado mais adiante.

Figura 12 - kNN segundo OpenCV.org



Fonte: (OPENCV.ORG (4), 2022)

Ainda segundo o autor anterior, na Figura 12, existem duas famílias: “caixas azuis” e “triângulos vermelhos”. Cada família é referenciada como uma Classe; suas posições são mostradas no mapeamento da imagem e é referenciado como espaço de recursos, que é como um espaço onde todos os dados são projetados. Se considerar um espaço de coordenadas 2D e

cada dado possui dois recursos, uma coordenada x e outra y . Podemos representar esse dado no seu espaço de coordenadas 2D. Se acrescentarmos mais um recurso, passará para o espaço em 3D, onde os recursos N são o espaço N -dimensional, que são o espaço de recursos. Na Figura 12, o espaço N -dimensional é 2D com dois recursos, se um novo membro surgir, um outro pixel, que é demonstrado como o círculo verde ao centro; devendo ser adicionado a uma dessas famílias azuis ou vermelhas ou melhor, em qual classe? Esse processo é conhecido como: Classificação.

O kNN, trabalha verificando quem é seu vizinho mais próximo e na Figura 12, fica claro que é um membro da família do triângulo vermelho e é classificado como um triângulo vermelho; assim, este método é conhecido como classificação do vizinho mais próximo, porque a classificação depende apenas do vizinho mais próximo.

O autor anterior ainda relata que há um problema com essa abordagem, pois o triângulo vermelho pode ser o vizinho mais próximo, mas e se também houver muitos quadrados azuis por perto, esses têm mais força na localização do que os triângulos vermelhos e apenas verificar que é o vizinho mais próximo não seria o suficiente, mas ao verificar alguns k famílias mais próximas. Na Figura 12, há $k = 3$, ou seja, considerando os 3 vizinhos mais próximos; o círculo verde tem dois vizinhos vermelhos e um vizinho azul (existem dois equidistantes do azul, mas como $k = 3$, considera-se apenas um), então assim ele deve ser adicionado à família vermelha. Mas e se tomarmos $k = 7$, ele possui 5 vizinhos azuis e 2 vizinhos vermelhos e deve ser adicionado à família azul. Portanto o resultado irá variar com o valor selecionado de k .

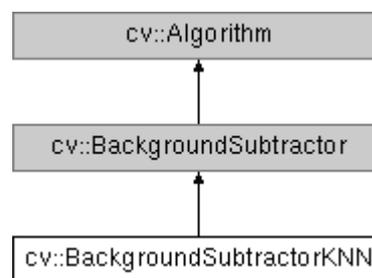
Observando que se k não for um número ímpar, irá ocorrer um empate, como aconteceria no caso acima com $k = 4$; pois tem 2 vizinhos vermelhos e 2 azuis como seus quatro vizinhos mais próximos e precisaríamos escolher um método para romper o empate para executar a classificação. Então, para reiterar, esse método é chamado k -Vizinho mais próximo uma vez que a classificação depende do k vizinhos mais próximos. Nesse caso ainda, os 2 vizinhos vermelhos estão realmente mais próximos do que os outros 2 vizinhos azuis, então ele é mais elegível para ser adicionado à família vermelha. A explicação matemática para essa resolução é dando pesos a cada vizinho, dependendo da distância do círculo verde, e assim aqueles que estão mais próximos dele têm pesos mais altos, enquanto aqueles que estão mais distantes obtêm pesos mais baixos. Em seguida é adicionados os pesos totais de cada família separadamente e o círculo verde é classificado como parte de qualquer família que tenha

recebido pesos totais mais altos; assim é chamado kNN modificado ou kNN ponderado; segundo (OPENCV.ORG (4), 2022)

A classe relacionada no Diagrama 5 e que implementa a subtração de fundo dos vizinhos mais próximos K descrito por (ZIVKOVIC e HEIJDEN, 2006) onde este algoritmo de segmentação de fundo calcula uma máscara de primeiro plano baseado em detecção de sombra, se verdadeiro, detecta sombras e as marca.

(HAMILTON *et al.*, 2018) descrevem o kNN como um aprendiz preguiçoso, que aprende por analogia, comparando um pixel desconhecido com seus pixels vizinhos mais próximos e cada pixel é descrito por atributos n , sendo a proximidade entre os pixels no espaço de decisão determinada pela similaridade entre seus atributos. A similaridade de atributo é definida em termos de uma métrica de distância, como a distância euclidiana. Afirmam que classificar um pixel desconhecido é conceitualmente simples para um classificador kNN; pois o pixel desconhecido está localizado no espaço de decisão, os k pixels mais próximos no espaço de decisão são localizados e assim o pixel desconhecido é atribuído à classe mais comum que ocorre entre os vizinhos mais próximos.

Diagrama 5 - Herança para *BackgroundSubtractorKNN*



Fonte: (OPENCV.ORG (4), 2022)

Assim como (HACK THE DEVELOPER, 2022) compara e exemplifica a utilização dos algoritmos kNN e BackgroundSubtractorMOG2.

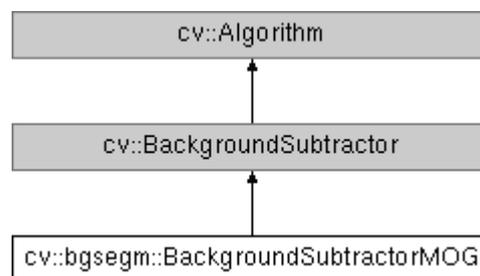
Os próximos algoritmos são similares para (GEEKSFORGEEKS, 2020) que os exemplifica na sua utilização prática, comparando os algoritmos: *BackgroundSubtractorMOG* (MOG), *BackgroundSubtractorMOG2* (MOG2) e *BackgroundSubtractorGMG* (GMG).

2.7.6 Algoritmo: Mixture of Gaussians (MoG)

Este algoritmo apresenta uma maior rapidez e precisão, além de se adaptar de maneira eficaz às mudanças do ambiente; pois conta com um esquema de detecção de sombra e é baseado em um espaço de cores computacional que faz uso de um modelo de plano de fundo conforme descrito por (KAEWTRAKULPONG e BOWDEN, 2002) e demonstrado no Diagrama 6. Ainda segundo os autores ele usa o método para modelar cada pixel de fundo por uma mistura de distribuições gaussianas K ($K = 3$ a 5); onde os pesos da mistura representam as proporções de tempo em que essas cores permanecem em cena e as cores prováveis do fundo são as que permanecem mais longas e mais estáticas.

BackgroundSubtractorMOG é um algoritmo de segmentação de *background/foreground* baseado em mistura gaussiana conforme descrito por (ZIVKOVIC, 2004) e (ZIVKOVIC e HEIJDEN, 2006).

Diagrama 6 - Herança para *BackgroundSubtractorMOG*



Fonte: (OPENCV.ORG (5), 2022)

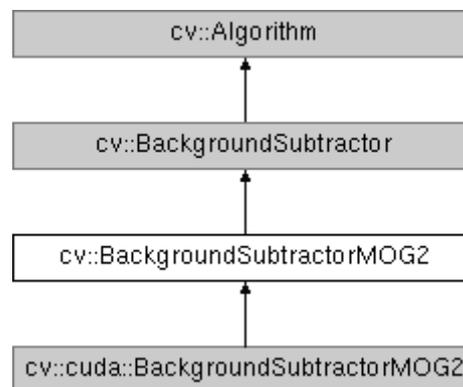
2.7.7 Algoritmo: Mixture of Gaussians version 2 (MoG2)

Algoritmo de segmentação de fundo, que calcula uma máscara de primeiro plano e implementa a subtração de fundo do modelo com uma mistura gaussiana proposta por (ZIVKOVIC, 2004) e posteriormente por (ZIVKOVIC e HEIJDEN, 2006), é representada no Diagrama 7. É também um algoritmo de segmentação de *background/foreground* baseado em mistura gaussiana, baseado nos dois artigos mencionados, e segundo os autores, sua característica mais importante é que ele seleciona o número apropriado de distribuição

gaussiana para cada pixel, que fornece uma melhor adaptabilidade a cenas variadas devido a alterações de iluminação etc.

Assim como (MARCOMINI e CUNHA, 2018) demonstram as diferenças e comparam o MOG, MOG2 e o GMG; ainda segundo os autores, que selecionaram cinco frames para cada um dos sete vídeos analisados, realizaram a análise em busca do levantamento de acurácia e precisão, além do tempo de processamento; concluindo que o MOG2 dentro das suas necessidades teve o melhor desempenho entre os algoritmos testados.

Diagrama 7 - Herança para *BackgroundSubtractorMOG2*



Fonte: (OPENCV.ORG (6), 2022)

2.7.8 Algoritmo: Godbehere-Matsukawa-Goldberg (GMG)

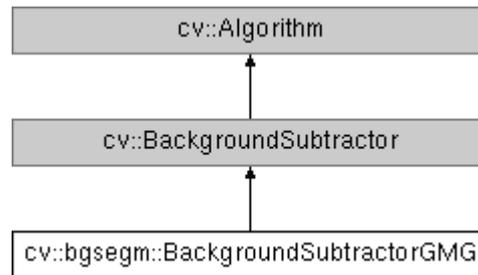
Esse algoritmo combina a estimativa estatística de imagem de fundo e a segmentação bayesiana por pixel. Foi introduzido por (GODBEHERE, MATSUKAWA e GOLDBERG, 2012) onde segundo os autores, o algoritmo usa os primeiros quadros (120 por padrão) para modelagem de fundo e emprega algoritmo probabilístico de segmentação em primeiro plano, identificando possíveis objetos em primeiro plano usando inferência bayesiana.

Ainda segundo os autores, estas estimativas são adaptáveis; as observações mais recentes, são mais pesadas do que as observações antigas, acomodando a iluminação variável; e várias operações de filtragem morfológica, como fechamento e abertura, são feitas para remover ruídos indesejados; por este motivo, recebe-se uma janela totalmente preta durante os primeiros quadros processados pelo algoritmo.

Em resumo, o módulo subtraidor de fundo baseado no algoritmo proposto por (GODBEHERE, MATSUKAWA e GOLDBERG, 2012) e demonstrado no Diagrama 8 e que

combina: estimativa estatística de imagem de fundo, segmentação *Bayesiana* por pixel e uma solução aproximada para o problema de rastreamento de múltiplos alvos usando um banco de filtros *Kalman* e correspondência *Gale-Shapley*.

Diagrama 8 - Herança para *BackgroundSubtractorGMG*



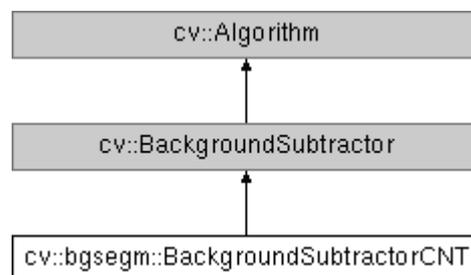
Fonte: (OPENCV.ORG (7), 2022)

2.7.9 Algoritmo: CouNT (CNT)

O algoritmo *BackgroundSubtractorCNT* é uma implementação do algoritmo Douglas-Peucker usado para aproximação de contorno no OpenCV, segundo (OPENCV.ORG (8), 2022) e o (LEARNOPENCV.COM, 2022) que detalham e exemplificam sua utilização.

Segundo o (OPENCV.ORG (8), 2022) este algoritmo de subtração de fundo com base na contagem, é tão rápido quanto o MOG2 em um sistema *high-end* e mais que o dobro, quando utilizado em um *hardware* como o *Raspberry Pi3* (por exemplo) e foi criado por Sagi Zeevi e disponibilizado em (ZEEVI, 2022), é demonstrada no Diagrama 9.

Diagrama 9 - Herança para *BackgroundSubtractorCNT*



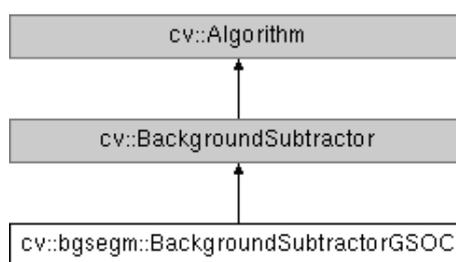
Fonte: (OPENCV.ORG (8), 2022)

2.7.10 Algoritmo: Google Summer of Code (GSoC)

Este algoritmo utiliza uma implementação diferente e é considerado ainda melhor, segundo o (OPENCV.ORG (9), 2022), pois foi implementado durante o *Google Summer of Code* (GSoC) e não foi originado de nenhum artigo, além de demonstrar um melhor desempenho no conjunto de dados CDNET 2014 (item 2.8 deste documento) em comparação com outros algoritmos no OpenCV e é demonstrado no Diagrama 10.

Segundo (MURZOVA, 2021) o GSoC foi fornecido como um avanço do LSPB, considerado mais rápido e robusto, dependente dos valores de cores RGB, o que difere do LSBP e atingiu um alto desempenho no CDNET-2012 e por não ser originário de nenhum artigo, sua análise mais aprofundada deve ser realizada em (SAMSONOV, 2020).

Diagrama 10 - Herança para *BackgroundSubtractorGSOc*



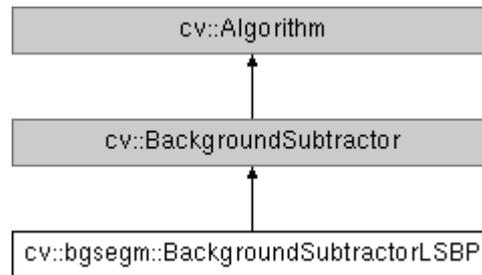
Fonte: (OPENCV.ORG (9), 2022)

2.7.11 Algoritmo: Local SVD Binary Pattern (LSBP)

Este algoritmo de Subtração em segundo plano utiliza o padrão binário local SVD. A maioria dos métodos de subtração de fundo depende do recurso de cor e textura; no entanto, devido a mudanças de iluminação em diferentes cenas e alterações de pixels de ruído, esses métodos muitas vezes resultaram em altos falsos positivos em um ambiente complexo; porém para resolver este problema, (GUO, XU e QIANG, 2016) propuseram um modelo de subtração de fundo adaptativo que usa um recurso de padrão binário local SVD (chamado LSBP) em vez de simplesmente depender da intensidade da cor. Esse recurso pode descrever a estrutura potencial das regiões locais em uma determinada imagem, assim, pode aumentar a robustez à variação de iluminação, ruído e sombras. O Diagrama 11 demonstra seu relacionamento.

Segundo (ROSEBROCK, 2015) este algoritmo é baseado no algoritmo *Local Binary Patterns* (LBP) que compara cada pixel com sua vizinhança de pixels circundante para construir uma representação local.

Diagrama 11 - Herança para *BackgroundSubtractorLSBP*



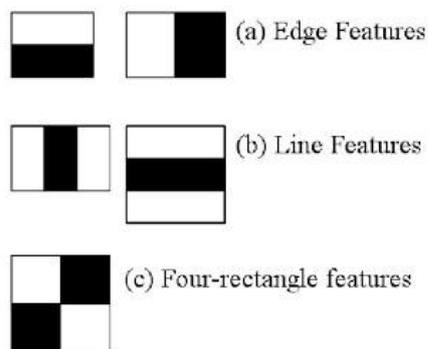
Fonte: (OPENCV.ORG (10), 2022)

2.7.12 Cascade Classifier, Haar Cascade e Algoritmo AdaBoost

Para (GEEKFORGEEKS, 2022) o Cascade Classifier é uma abordagem baseada no aprendizado de máquina usada para detectar objetos em imagens ou vídeos e combina classificadores cada vez mais complexos, como o AdaBoost, em cascata, o que permite entrada negativa.

Segundo o (OPENCV.ORG (11), 2022) para a detecção de objetos usando classificadores em cascata baseados em recursos Haar é um método eficaz de detecção de objetos proposto (VIOLA e JONES, 2001); sendo uma abordagem baseada no aprendizado de máquina, onde uma função em cascata é treinada a partir de muitas imagens positivas e negativas; usado para detectar objetos em outras imagens. Inicialmente, o algoritmo precisa de muitas imagens positivas (imagens do objeto que se deseja identificar em diferentes condições e opções, quando aplicável) e imagens negativas (imagens sem os objetos de interesse) para treinar o classificador; após, pode-se extrair recursos dele e para isso, são utilizados os recursos de Haar mostrados na Figura 13 que agem como *kernel* convolutivo e cada recurso tem um valor único, obtido da subtração do valor da soma de pixels sob o retângulo branco e a soma dos pixels sob o retângulo preto.

Figura 13 - Haar



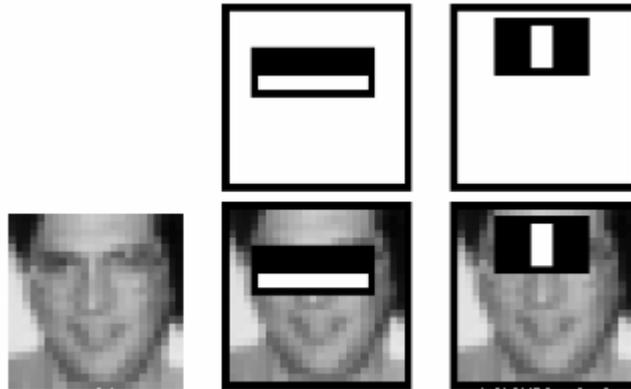
Fonte: (OPENCV.ORG (11), 2022)

Ainda segundo (OPENCV.ORG (11), 2022), todos os tamanhos e locais possíveis de cada *kernel* são usados para calcular os recursos, por exemplo; uma janela 24x24 resulta em mais de 160.000 recursos e para cada cálculo de recurso, é preciso encontrar a soma dos pixels em retângulos branco e preto; para resolver isso, eles introduziram a imagem integral. Por maior que uma imagem seja, ela reduz os cálculos para um determinado pixel, uma operação que envolve apenas quatro pixels, e segundo (VIOLA e JONES, 2001) isso torna as coisas mais rápidas. Na Figura 14, a linha superior mostra dois recursos; o primeiro selecionado parece focar na propriedade de que a região dos olhos é frequentemente mais escura que a região do nariz e das bochechas. O segundo, depende da propriedade de que os olhos são mais escuros que a ponte do nariz. Mas as mesmas janelas aplicadas nas bochechas ou em qualquer outro lugar são irrelevantes; a seleção dos melhores recursos entre os +160.000 é alcançado por AdaBoost.

Segundo (OPENCV.ORG (D), 2021) descreve a aplicação de todos os recursos em todas as imagens de treinamento e para cada recurso, o *AdaBoost* encontra o melhor limite que classificará as faces como positivas e negativas. Obviamente, haverá erros ou erros de classificação, para solucionar isso, é selecionado os recursos com taxa de erro mínima, o que significa que eles são os recursos que classificam com mais precisão as imagens de rosto e o que não é de rosto; onde cada imagem recebe um peso igual no início. Após cada classificação, os pesos das imagens mal classificadas são aumentados; onde o mesmo processo é feito e novas taxas de erro são calculadas; gerando novos pesos; então o processo continua até que a precisão ou taxa de erro necessária seja alcançada ou o número necessário de recursos seja encontrado; o classificador final é uma soma ponderada desses classificadores fracos; sendo chamado de

fraco, porque por si só não pode classificar a imagem, mas em conjunto com outros; forma um forte classificador.

Figura 14 - AdaBoost



Fonte: (OPENCV.ORG (D), 2021)

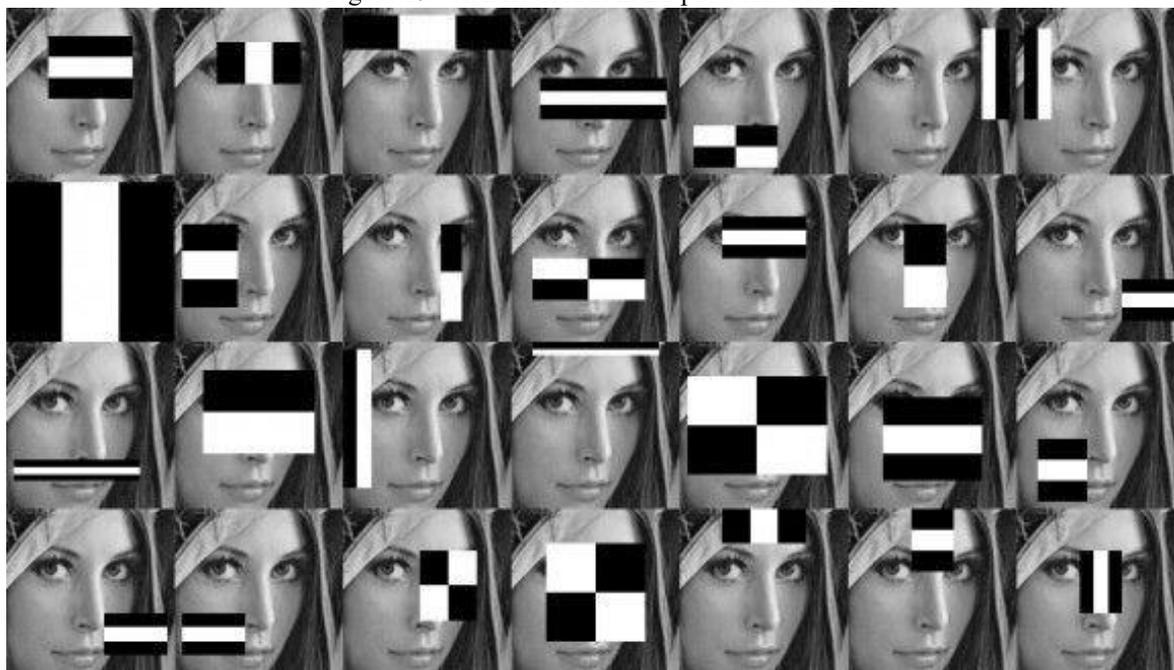
(VIOLA e JONES, 2001) afirmam que até 200 recursos fornecem detecção com 95% de precisão e sua configuração final tinha cerca de 6000 recursos; uma redução de +160.000 para 6.000 recursos. Porém ainda não satisfatório no quesito de eficiência, em uma imagem, a maior parte da imagem é de região não facial. Portanto, é uma ideia melhor ter um método simples para verificar se uma janela não é uma região de face. Caso contrário, descarte-o em uma única foto e não o processe novamente. Em vez disso, concentre-se em regiões onde pode haver um rosto. Dessa forma, passamos mais tempo verificando possíveis regiões de face. Para isso, eles introduziram o conceito de *Cascade of Classifiers*. Em vez de aplicar todos os 6.000 recursos em uma janela, os recursos são agrupados em diferentes estágios dos classificadores e aplicados um por um. Assim Normalmente, os primeiros estágios conterão muito menos recursos e se uma janela falhar no primeiro estágio, ela é descartada e não é considerado os recursos restantes. Se passar, é aplicada o segundo estágio dos recursos e o processo é continuado; a janela que passa por todas as etapas é uma região de face.

O detector de (VIOLA e JONES, 2001) tinha mais de seis mil recursos, com trinta e oito estágios, com 1, 10, 25, 25 e 50 recursos nas cinco primeiras etapas. Os dois recursos na Figura 14 são obtidos como os dois melhores recursos do *AdaBoost* e segundo os autores, em média dez recursos de mais de seis mil foram avaliados por sub janela.

Segundo (JAISWAL, 2022), para identificar um objeto personalizado em uma imagem é conhecido como detecção de objeto, tarefa que pode ser feita usando várias técnicas, mas o

método mais simples para executar a detecção de objetos é o proposto por (VIOLA e JONES, 2001), como demonstra a Figura 15 e a Figura 16.

Figura 15 - HAAR CASCADES por Viola e Jones

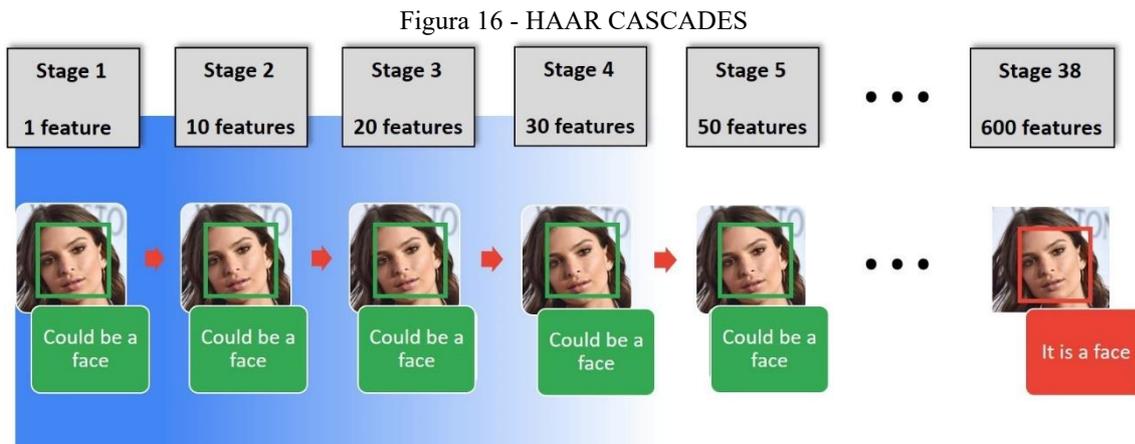


Fonte: (MITTAL, 2020)

Mas segundo (JAISWAL, 2022) há limitações para o processo, como:

- Alta detecção de falso positivo
- Menos preciso do que as técnicas baseadas em aprendizado profundo
- Ajuste manual de parâmetros.
- Treinar *Haar Cascades* em um objeto personalizado não é fácil.

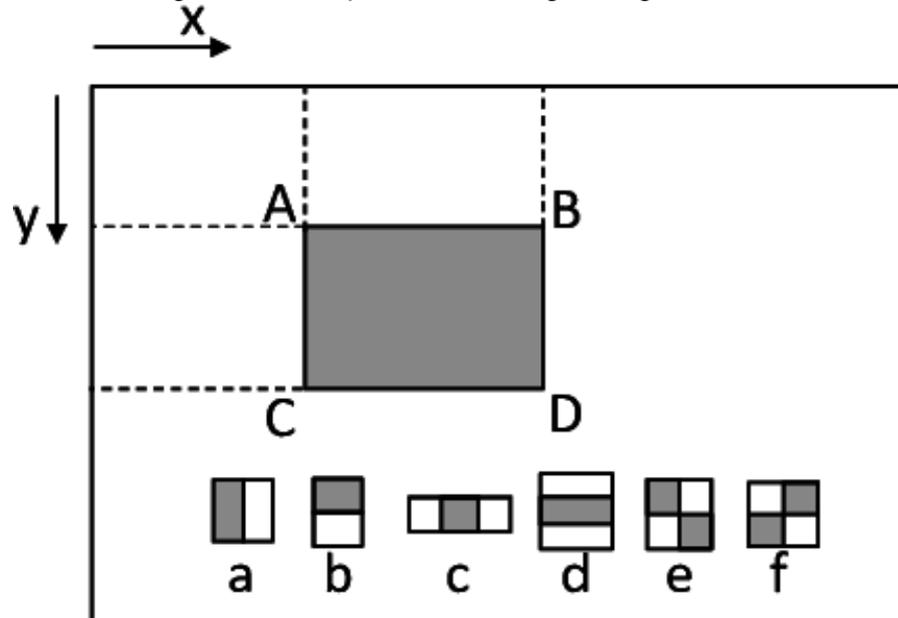
O autor anterior ainda conclui que mesmo sendo amplamente utilizado quando se pode aceitar alguma detecção de falso positivo, o *Haar Cascade* deve ter algum ajuste manual do parâmetro de detecção, que pode resolver o problema da detecção de falso positivo, recomenda a utilização de outro algoritmo de detecção de objetos, como o *YOLO* (uma ferramenta de visão computacional, acrônimo de “*You Only Look Once*”), promete melhor precisão com um bom *frames per second* (FPS); além de treinar uma *Haar Cascade* para um objeto personalizado não é muito eficiente.



Fonte: (JAISWAL, 2022)

Para (MITTAL, 2020) as imagens integrais aceleram essencialmente o cálculo dos recursos do *Haar* e em vez de calcular a cada pixel, ele cria sub retângulos e cria referências de matriz para cada um desses sub retângulos, que são usados para calcular os recursos do *Haar*, como demonstra a Figura 17; o autor afirma que é importante observar que quase todos os recursos do *Haar* serão irrelevantes ao fazer a detecção de objetos, porque os únicos recursos importantes são os do objeto; mas ao determinar os melhores recursos que representam um objeto das centenas de milhares de recursos do *Haar*, o *AdaBoost* escolhe essencialmente os melhores recursos e treina os classificadores para usá-los, usando uma combinação de classificadores fracos e assim criar um classificador forte, que o algoritmo pode usar para detectar objetos.

Figura 17 - Ilustração de como a imagem integral funciona



Fonte: (MITTAL, 2020)

2.8 BENCHMARK CDNET 2014

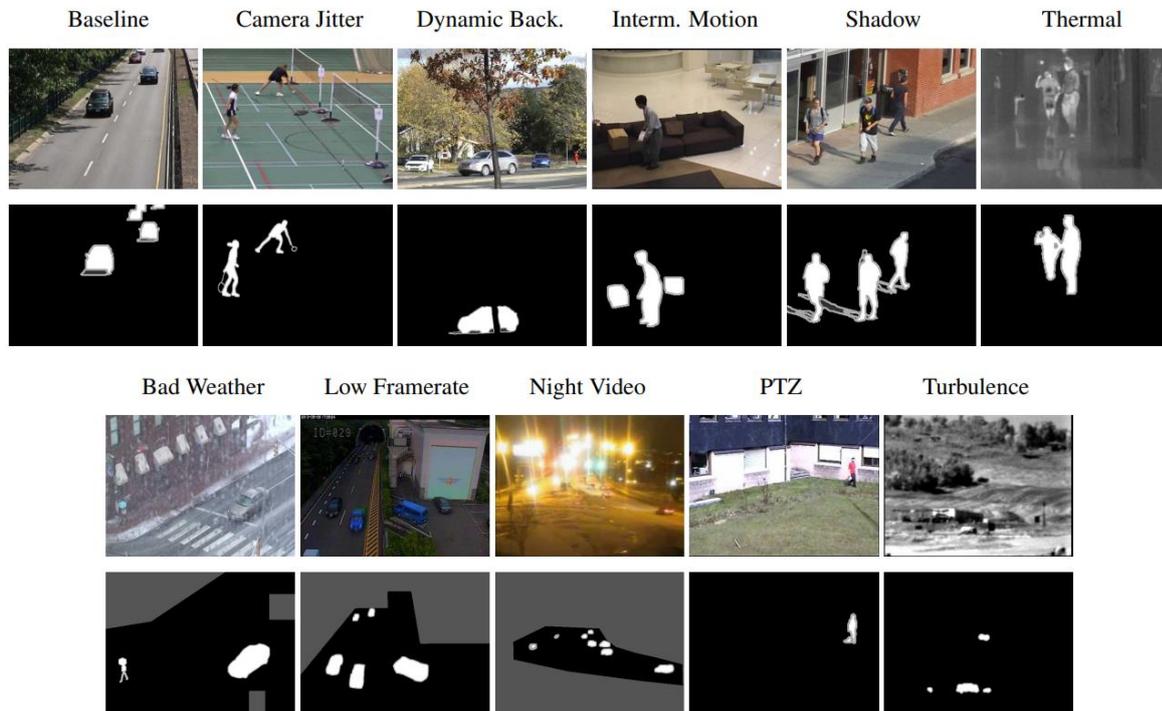
CDnet2014 é um banco de dados de vídeo para testar algoritmos de detecção de alterações, segundo (OUSSAMA, 2019).

Sua introdução em 2012 evoluiu em 2014 e é considerada uma versão expandida do conjunto de dados originais, segundo (CHANGEDETECTION.NET, 2015) e seu conjunto de dados encapsula um esforço de *benchmarking* acadêmico rigoroso e abrangente para testar e classificar algoritmos existentes e os novos.

É usado para avaliar o desempenho de algoritmos em vários cenários, como alterações de iluminação, movimento da câmera, objetos dinâmicos etc. (WANG *et al.*, 2014)

A detecção de alterações é uma das tarefas de baixo nível mais importantes na análise de vídeo. Em 2012, (WANG *et al.*, 2014) introduziram o *benchmark changedetection.net* (CDnet), um conjunto de dados de vídeo dedicado à avaliação de abordagens de mudança e detecção de movimento, conforme demonstra a Imagem 4.

Imagem 4 - Exemplos de quadros de vídeo de cada uma das 11 categorias no conjunto de dados CDNET 2014



Fonte: (WANG *et al.*, 2014)

A44

Apresentando uma versão do conjunto de dados *CDnet*, que incluí 22 vídeos adicionais (70.000 quadros anotados em pixels) abrangendo 5 novas categorias que incorporavam desafios encontrados em muitos ambientes de vigilância. Ainda descreveram essas categorias em detalhes e forneceram uma visão geral dos resultados de mais de uma dúzia de métodos submetidos ao “*IEEE Change Detection Workshop 2014*”. Além de destacarem os pontos fortes e fracos desses métodos e identificaram os problemas remanescentes na detecção de mudanças. Na Imagem 4 encontra-se exemplos de quadros de vídeos de cada uma das 11 categorias no conjunto de dados disponibilizados em www.changedetection.net. Sendo que as categorias na primeira linha existem nos conjuntos de dados de 2012 e 2014, e as da segunda linha, existem apenas no conjunto de dados de 2014.

2.9 TRABALHOS RELACIONADOS

A visão computacional, segundo (IFTIKHAR, DISSANAYAKE e VIAL, 2014) é uma tecnologia que usa algoritmos para interpretar e entender dados visuais, como imagens e vídeos e na Figura 18 exemplifica as etapas do processo.

Figura 18 - Demonstração: (a) Quadro de vídeo original, (b) Primeiro plano detectado, (c) Operação Morfológica, (d) Veículos Detectados



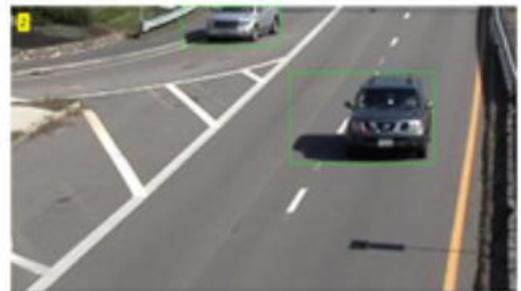
(a)



(b)



(c)



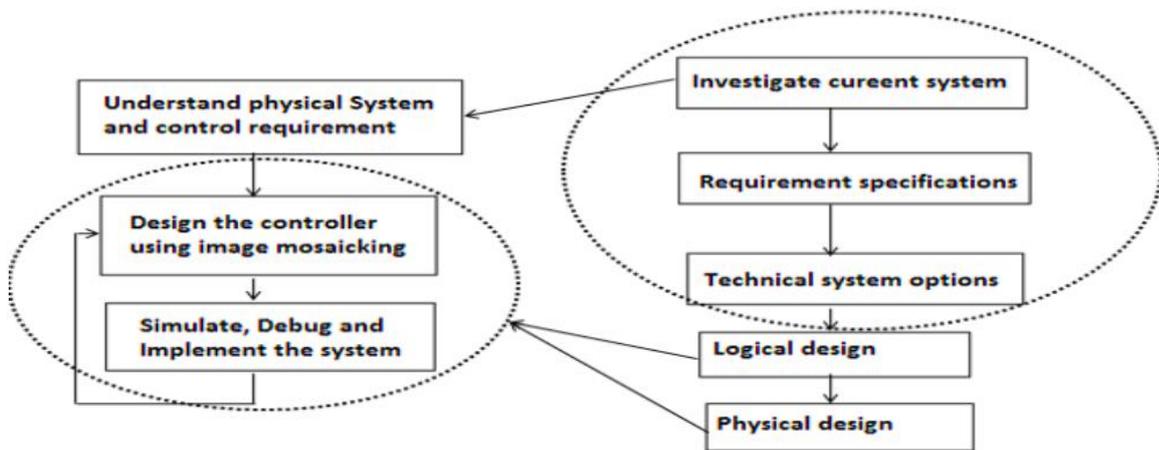
(d)

Fonte: (IFTIKHAR, DISSANAYAKE e VIAL, 2014, p. 345)

A tecnologia de redes neurais artificiais, que em ciência da computação e campos relacionados são modelos computacionais inspirados pelo sistema nervoso central de um animal e que são capazes de realizar o aprendizado de máquina, bem como o reconhecimento de padrões onde novos resultados são previstos com base na experiência anterior. No entanto, muitos sistemas híbridos também são desenvolvidos para ambientes urbanos, como o controle de sinal de tráfego apresentado inicialmente por (PATEL e RANGANATHAN, 2001).

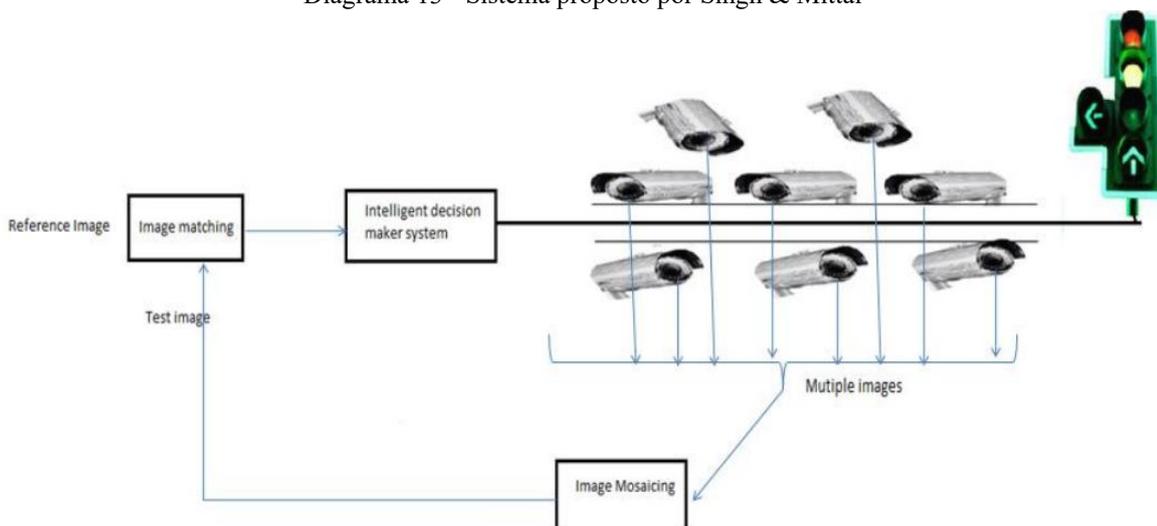
(SINGH e MITTAL, 2013) propõem resolver o problema de congestionamento de tráfego por meio de uma análise e projeto de sistema inteligente, conforme a metodologia exemplificada no Diagrama 12 e no Diagrama 13 onde é apresentando um algoritmo de processamento de imagem simples, para estimar a densidade de tráfego em um ponto de acesso, usando feeds de câmeras de circuito fechado de televisão (CFTV). Com base na comparação de imagens, após a integração das imagens de várias projeções, um algoritmo de tomada de decisão inteligente, foi capaz de controlar o congestionamento.

Diagrama 12 - Metodologia proposta por Singh & Mittal



Fonte: (SINGH e MITTAL, 2013, p. 4)

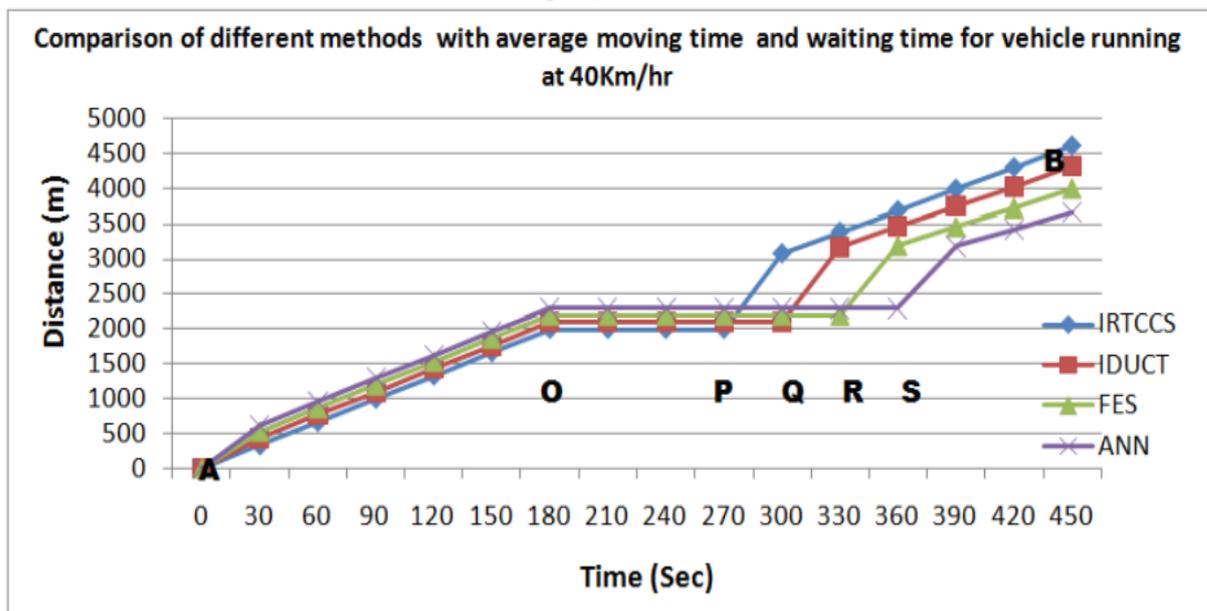
Diagrama 13 - Sistema proposto por Singh & Mittal



Fonte: : (SINGH e MITTAL, 2013, p. 4)

Na análise e projeto propostos por (MITTAL, SINGH e SHARMA, 2015), onde um algoritmo para a medição do desempenho por meio do rastreamento da densidade do tráfego rodoviário empregado no modelo proposto, e que os erros de estimativa do tempo de viagem podem ser autocorrigidos com os dados de status do sinal, porque as diferenças de correspondência entre uma imagem atual e uma imagem atualizada decidem os tempos dos semáforos. A eficácia foi avaliada, utilizando os softwares *MATLAB*, *JAVA* e *LabVIEW*. Com base na medição dinâmica precisa da densidade do tráfego, a técnica dinâmica para gerenciar os semáforos foi desenvolvida com o objetivo de maximizar o tráfego e minimizar o tempo médio de espera em uma interseção. Na opinião dos autores, algumas técnicas dinâmicas artificiais como lógica *Fuzzy*, *Fuzzy Expert Systems* (FES), *Artificial Neural Network* (ANN), entre outras; podem ser desenvolvidas com mosaicos de imagens, para melhorar a decisão em menor tempo com base em medidas de densidade precisa. Conforme demonstra o Gráfico 3.

Gráfico 3 - Comparação de técnicas DRTCC



Fonte: (MITTAL, SINGH e SHARMA, 2015, p. 8)

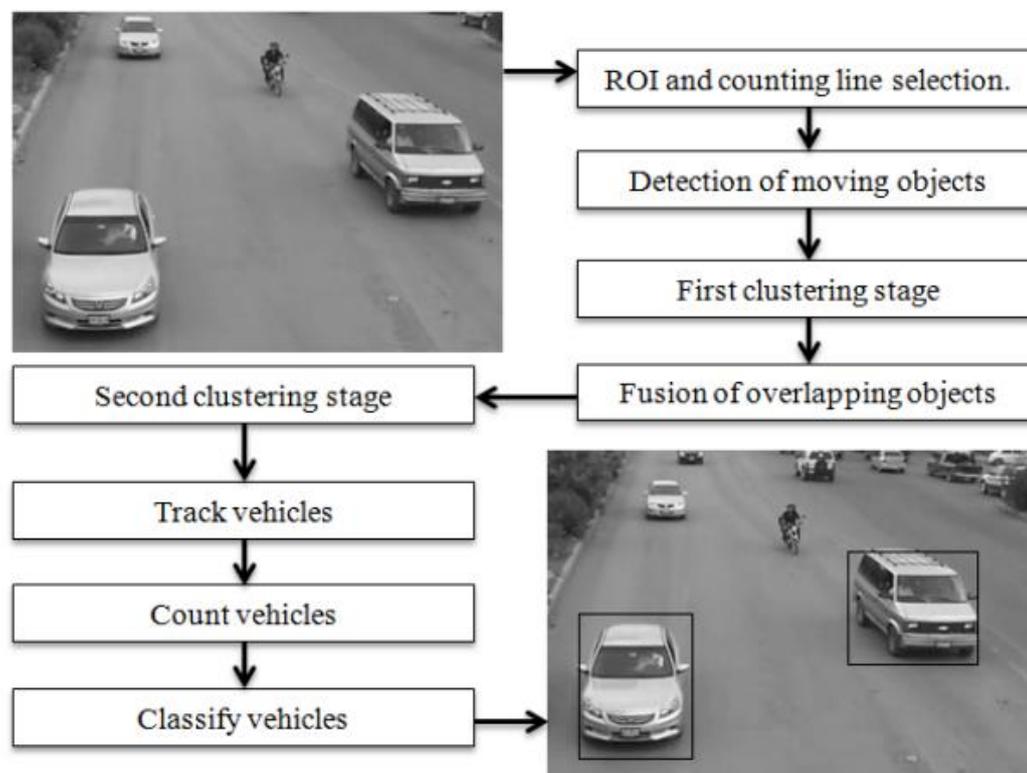
Para (HAWI, OKEYO e KIMWELE, 2015), que revisaram sobre o uso de tecnologia para controlar e gerenciar o tráfego, observando a implementação de tecnologia inteligente em sistemas de transporte, que tem um impacto substancial nos níveis de tráfego. Comparando os sistemas estáticos com os que tem base em IA e embora estes sistemas forneçam benefícios

substanciais para a gestão do tráfego, FES e ANN são um ramo da IA que ainda é um campo emergente; daí a implementação de tais sistemas se torna bastante caro, uma alternativa apresentada é o STCS para usar os sistemas IA em redes de sensores, devido ao baixo custo de implementação em comparação com os sistemas baseados em IA. A rede é composta por muitos sensores que cooperam para monitorar e coletar dados sobre o tráfego; onde essas informações são então encaminhadas para um controlador que processa os dados. Usando um algoritmo, o controlador é capaz de fazer decisões de roteamento com base nas condições de tráfego. Segundo os seus autores, os sistemas de controle de tráfego inteligentes ainda possuem limitações para o que eles podem alcançar de forma inteligente, o futuro é promissor para esses sistemas e que os sistemas inteligentes são o caminho a seguir para o controle do tráfego.

Para (TAN, SULLIVAN e BAKER, 1994), que apresentam um algoritmo para a localização e reconhecimento de veículos em cenas de tráfego que, em condições normais, são obrigadas a estar em contato com o plano-terra; esse algoritmo elimina a necessidade de extração e correspondência explícita de recursos e reduz substancialmente o custo computacional. Esse algoritmo foi testado extensivamente com imagens de tráfego externo com resultados bem-sucedidos, e que ainda, foi desenvolvido para implementação em tempo real para aplicações em análise de tráfego e/ou pode ser adaptado a outras aplicações industriais, onde os objetos estão em contato com um plano conhecido.

Para (PEÑA-GONZÁLEZ e NUÑO-MAGANDA, 2014), que abordam um plano que detecta veículos em tempo real e que é possível contar o número de veículos e classificar cada veículo com a ajuda do sistema de visão computacional. Uma câmera de alta definição é instalada para obter a filmagem em tempo real e ao mesmo tempo, diferentes algoritmos de classificação e agrupamento, são usados para processar os mesmos dados da filmagem. O método proposto, como demonstrado na Figura 19, divide a imagem de entrada em uma grade retangular de $N \times N$. Esta abordagem, alcançou eficiência de mais de 95% nos casos de teste experimental e seus sistemas também obtêm uma taxa de processamento de até 30 FPS em imagens com resolução de 1280 x 720 pixels.

Figura 19 – Sistema de detecção de veículos proposto por PEÑA-GONZÁLEZ e NUÑO-MAGANDA, 2014

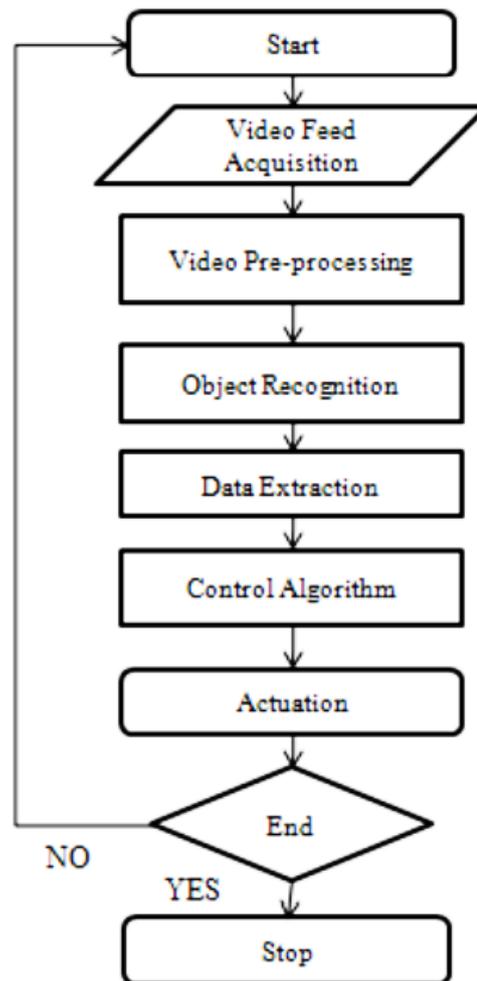


Fonte: (PEÑA-GONZÁLEZ e NUÑO-MAGANDA, 2014, p. 680)

(BHASKAR e YONG, 2014) aplicaram um método baseado em processamento de imagem para rastrear e identificar os veículos em condições de tráfego misto em condições de baixo, médio e alto tráfego com precisão, além de contar os veículos. A limitação do método desenvolvido, que utiliza detecção de BLOB e modelo de mistura gaussiana, é que exige ajustes dos parâmetros para obter o melhor desempenho em condições de tráfego intenso, onde requer mais tempo de processamento.

Os resultados obtidos por (ALI *et al.*, 2015) sugerem que o uso de extração de recursos de vários feeds de vídeo em combinação com o algoritmo de controle proposto resulta em melhoria de desempenho significativo em comparação com o controlador de tempo fixo. Como demonstrado no Fluxograma 4, Tabela 3 e Fluxograma 5.

Fluxograma 4 – Etapas envolvidas na resolução de congestionamento de tráfego



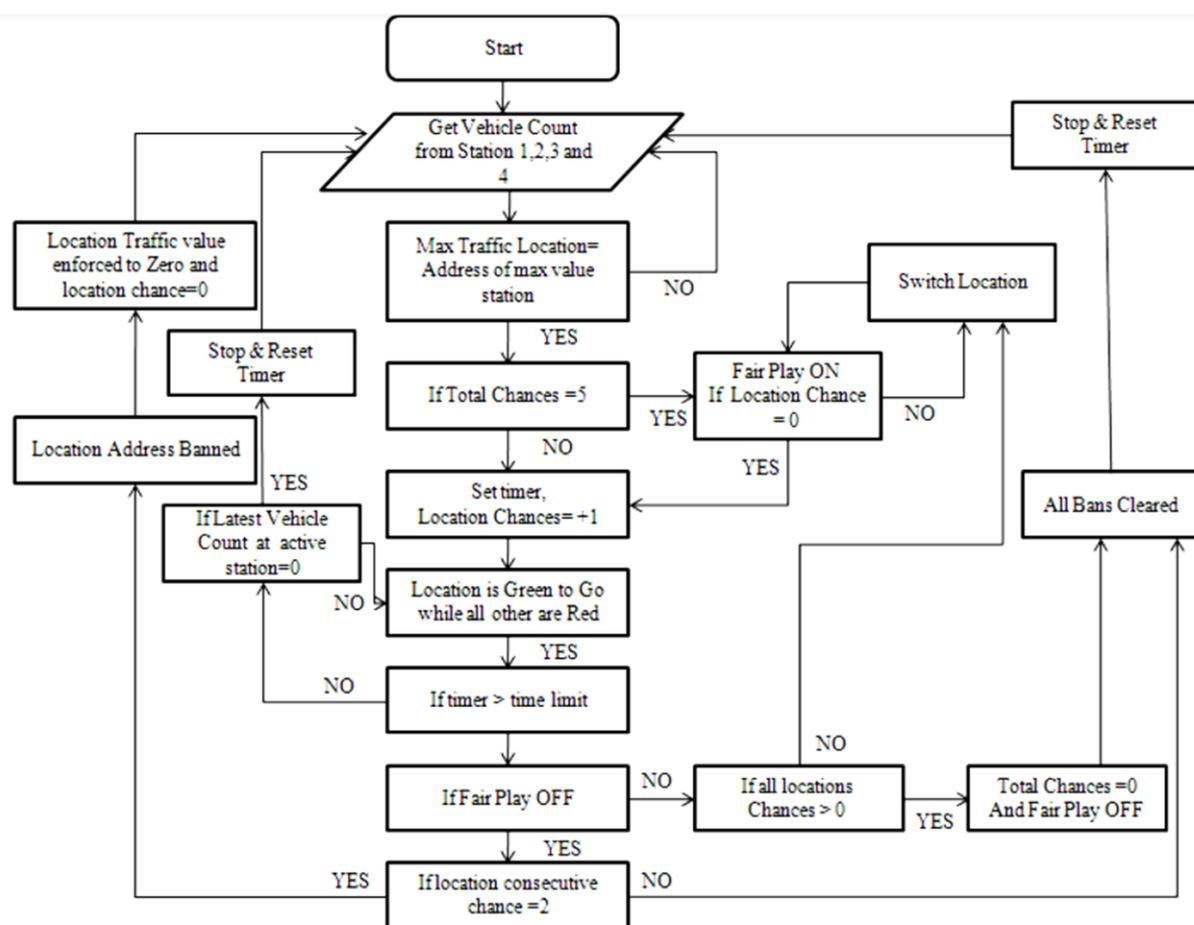
Fonte: (ALI *et al.*, 2015, p. 229)

Tabela 3 – Resultados de performance do classificador em fase de teste

Tipo de Veículo	Detecções corretas	Detecções falhas	Precisão
Carros	98	2	98%
Caminhões	71	9	88,75%
Ônibus	82	3	96,4%
Furgões	37	3	92,5%

Fonte: Traduzido pelo autor, do original de (ALI *et al.*, 2015)

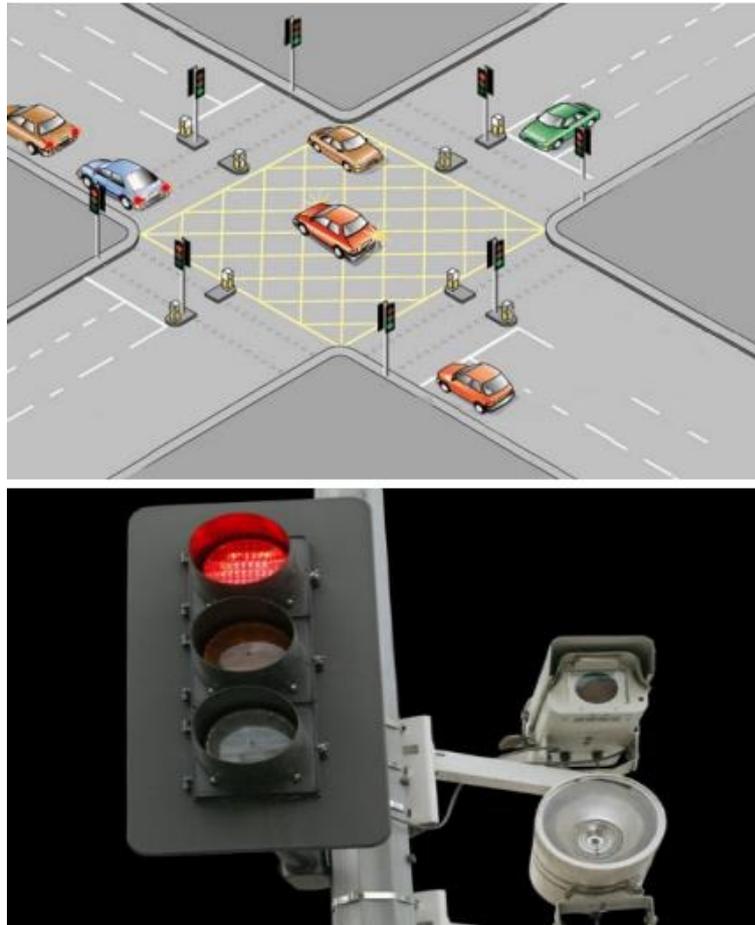
Fluxograma 5 - Algoritmo de controle proposto por ALI, MALIK, et al., 2015



Fonte: (ALI *et al.*, 2015, p. 230)

(KANUNGO, SHARMA e SINGLA, 2014) trabalharam com o sistema utilizando câmeras de vídeo nos cruzamentos de tráfego de cada lado da via, como se fosse um cruzamento de quatro vias. Portanto, quatro câmeras de vídeo foram instaladas sobre as luzes vermelhas voltadas para a via, para mais detalhes veja a Figura 20. As câmeras capturam o vídeo e o transmitem para os servidores onde, usando técnicas de processamento de vídeo e imagem, a densidade do veículo em cada lado da via é calculada e um algoritmo é empregado para mudar os semáforos. Assim o hardware também inclui a conexão dessas câmeras ao servidor para receber transmissão ao vivo e um servidor com capacidade suficiente para lidar com os requisitos de processamento, o software usado no sistema inclui o *MATLAB* com *Image Processing and Video Processing Toolbox* e um compilador C++ para gerar resultados algorítmicos.

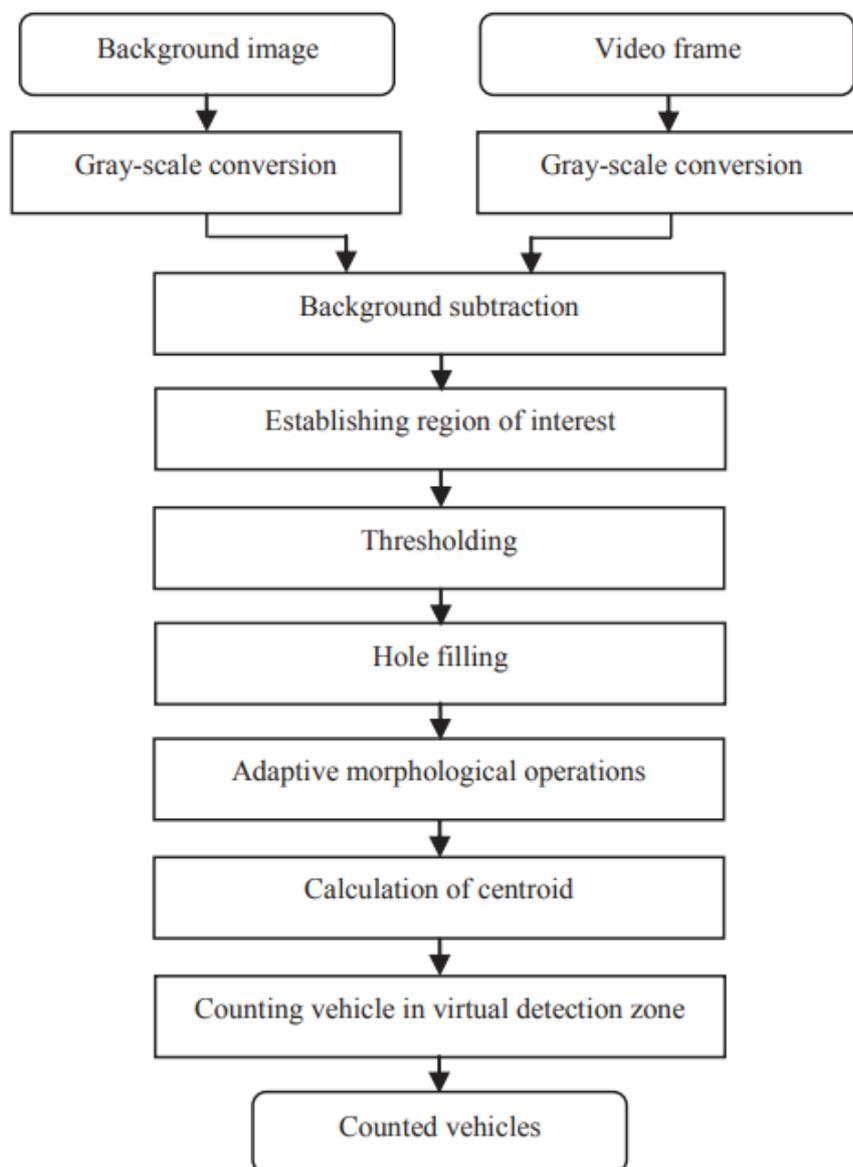
Figura 20 – Layout proposto por KANUNGO, SHARMA e SINGLA, 2014



Fonte: Adaptado de (KANUNGO, SHARMA e SINGLA, 2014)

(SEENOUVONG *et al.*, 2016) descrevem um sistema de detecção e contagem de veículos que desempenha um papel importante em um sistema de transporte inteligente, especialmente para o gerenciamento de tráfego, que aborda um método baseado em detecção e contagem de veículos por vídeo e visão computacional. No Fluxograma 6, o método proposto usa técnica de subtração de fundo para encontrar objetos de primeiro plano em uma sequência de vídeo, com isso detectar mais veículos em movimento com precisão, várias técnicas de visão computacional, incluindo operações de limiarização, preenchimento de orifícios e morfologia adaptativa, são então aplicados. Finalmente, a contagem de veículos é feita usando uma zona de detecção virtual e conseguindo assim resultados experimentais com precisão do sistema de contagem de veículos em torno de 96%

Fluxograma 6 - Método proposto por (SEENOUVONG, WATCHAREERUETAI, et al., 2016)



Fonte: (SEENOUVONG *et al.*, 2016, p. 225)

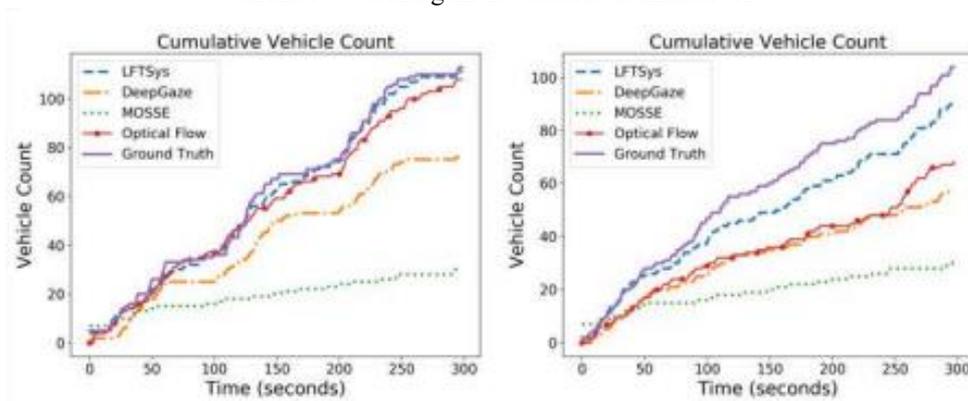
(WEI *et al.*, 2019) descrevem como o rastreamento de veículos e a estimativa do fluxo de tráfego são problemas que têm muitas aplicações em transporte inteligente e cidades inteligentes e como que o fluxo de veículos desempenham papéis importantes em sistemas de transporte inteligentes (ITS), fornecendo informações de tráfego em tempo real, para que seja analisado ou que se necessário, medidas sejam tomadas para prevenção de congestionamentos, planejamento de rotas e ainda, que sejam coletadas informações históricas de longo prazo para serem utilizadas no planejamento urbano. Como demonstra a Imagem 5 e o Gráfico 4.

Imagem 5 - Veículos em condições ambientais urbanas e rodoviárias



Fonte: (WEI *et al.*, 2019, p. 609)

Gráfico 4 – Contagem de veículos acumulativos.



Fonte: (WEI *et al.*, 2019, p. 609)

(GREESHMA *et al.*, 2019) utilizando de OpenCV em um sistema proposto introduziram um sistema de controle de tráfego baseado na densidade de veículos para evitar os problemas de congestionamento, proporcionando fluidez ao tráfego. Conseguiram controlando a densidade do tráfego nas estradas, afirmando que o sistema introduziu um novo método para

controlar a densidade de veículos controlando os semáforos usando processamento de imagem, e que a densidade de veículos é medida usando classificadores predefinidos e disponíveis no processamento de imagens. Controlando através da densidade de veículos, que se estiver acima do valor limite, fornece ao equipamento de gerenciamento semafórico, o sinal de tráfego apropriado para a via monitorada.

Além disso (WEI *et al.*, 2019) defendem que o rastreamento pode ser usado para novas informações em tempo real em toda a cidade com aplicações para pedestres e veículos; ainda descrevem como muitas cidades implantaram redes de câmeras em cruzamentos com a finalidade de monitorar e analisar o tráfego e fornecer dados para grupos interessados; mas que acabam sendo limitados as limitações de largura de banda e armazenamento, os fluxos de vídeo em resolução FullHD em tempo real geralmente não estão disponíveis.

Os autores ainda citam como exemplo, o Departamento de Transportes da cidade de Nova York que disponibilizou ao público feeds de vídeo das 752 câmeras de trânsito, mas que ainda são transmitidos com uma taxa de quadros extremamente baixa e em uma resolução de 352x240 pixels. Uma vez que muitos métodos atuais para rastreamento de veículos dependem de imagens de alta taxa de quadros e alta resolução, essas fontes de dados levam a altas taxas de erro para os métodos atuais e não são utilizáveis.

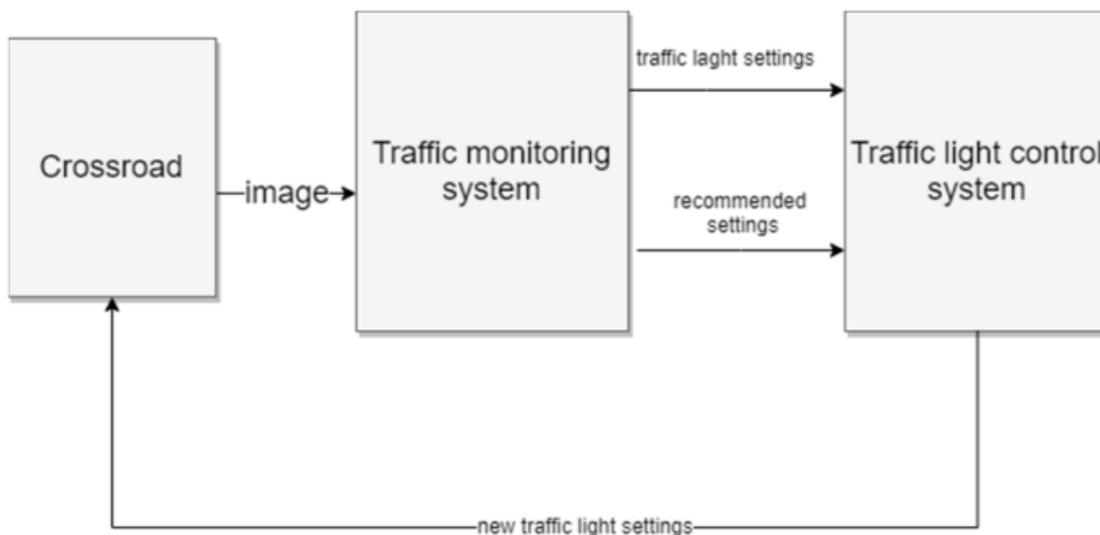
Segundo ainda os autores, que apresentam um sistema de rastreamento de veículos e de fluxo de tráfego em tempo real, o *Low Frame-rate Tracking System (LFTSys)*, que utiliza vídeos com baixa taxa de quadros e que utiliza câmeras de tráfego existentes e podendo ser implantado em escala, fornecendo assim, segundo os autores, benefícios imediatos para cidades, pedestres e motoristas, por ser adaptável a diferentes ambientes e perspectivas de câmera, assim conseguindo ser flexível a diferentes partes da cidade e com implementação quase que imediata em cidades com redes de câmeras com baixa taxa de quadros, como Nova York, Los Angeles e Austin.

Como (CAVALHEIRO, 2021) expõe em seu trabalho direcionado à cidade de São Paulo e em relação ao uso de câmeras, evidenciando a existência de focos distintos na implementação de sistemas inteligentes de gestão de tráfego, desde a sua utilização e análise de imagens para agilizar a travessia de pedestres, onde o foco não é a otimização do tráfego, mas a segurança dos pedestres, focando na redução de acidentes na área de cruzamentos. Ou ainda a convergência no uso dos sistemas em tempo real com detecção de imagens para tornar a gestão do tráfego mais eficiente, conseqüentemente com isso, a redução dos congestionamentos. Mencionando as imagens que são capturadas e analisadas por meio de algoritmos de

processamento de imagem para adequar a temporização dos semáforos a demanda em *real-time*.

(ZINCHENKO *et al.*, 2020) apresentam as abordagens e os métodos em visão computacional para a detecção de objetos em movimento e seu rastreamento e investigam formas de usar métodos de visão computacional para otimizar e controlar o tráfego. Afirmam que o objetivo no desenvolvimento de sistemas modernos de gerenciamento de tráfego é criar mecanismos eficazes de gerenciamento de tráfego de acordo com as condições dinâmicas do tráfego e que os atuais sistemas que regulam o trânsito apresentam muitos inconvenientes, entre eles que os sistemas trabalham de acordo com um programa predefinido e que esses não trabalham com dados em tempo real. O trabalho destes autores se concentra em uma nova abordagem, onde o gerenciamento do tráfego rodoviário, incorpora um sistema inteligente de controle de semáforos usando um algoritmo que consome dados reais das câmeras de circuito fechado de televisão (CCTV) e arte da solução, foi desenvolvida com o emprego de um programa utilizando uma plataforma de programação popular que calcula as ordens de acionamento para a rede de semáforos; sua proposta de melhorar os resultados em termos de redução do tempo de espera de pedestres e veículos, menor tempo de viagem e aumento da velocidade média dos veículos, conforme a Figura 21.

Figura 21 - Sistema proposto por (ZINCHENKO, KONDRATENKO, et al., 2020)



Fonte: (ZINCHENKO *et al.*, 2020, p. 252)

Segundo (JIMÉNEZ-BRAVO *et al.*, 2022) a área de pesquisa em visão computacional possui diferentes subáreas que tiveram papéis fundamentais ao longo do tempo, como exemplo, citam a classificação de imagens, área que consiste em identificar o objeto ou objetos que aparecem na imagem e desta forma, na saída do processo, será o objeto que irá compor a imagem ou uma classificação. Mas segundo os autores, a classificação só é capaz de prever quais elementos estão presentes na imagem e não onde estão, então se justifica a existência de técnicas de detecção de objetos; estas detectam a posição exata dos objetos em uma imagem e ainda são capazes de detectar uma quantidade muito grande de tipos de objetos que estejam sendo considerados. Consideram que a detecção de objetos foi uma grande revolução na área de visão computacional e são considerados um dos pilares fundamentais nas técnicas de rastreamento de imagens.

Para (PEREZ, 2021) as redes de transporte são infraestruturas consideradas críticas para a sociedade em centros urbanos onde a malha formada por ruas e avenidas, possibilitam o transporte de pessoas e produtos, além da realização de serviços essenciais da área de saúde, segurança pública, comércio, lazer, entre outros. Afirmando ainda, que os sistemas de transporte são sistemas complexos e compostos por grande variedade de agentes e estrutura, que interagem localmente e desta forma, geram os padrões globais, dando como exemplo: o desenvolvimento econômico urbano. Exemplifica ainda os semáforos e outros dispositivos de controle de fluxo, inteligentes e integrados à tecnologia da informação; que vem se tornando cada vez mais comuns, mas faz um alerta; pois estes dispositivos não são a prova de falhas e possuem vulnerabilidades aos ataques cibernéticos ou ainda as falhas de infraestrutura, como a perda de conectividade (comunicação) e as panes elétricas, o que certamente afeta a integridade de todo o sistema. Esse impacto disruptivo de falhas ou ataques, está relacionado à forma como os sistemas são estruturados.

O autor anterior ainda teoriza sobre as redes complexas e como isso fornece ferramentas, consideradas importantes para a análise de redundância e robustez de sistemas complexos, como o sistema de tráfego. Considerando a relação entre falhas nas interseções e a estrutura de sistemas de transporte, busca com o auxílio da teoria de redes complexas, avaliar o impacto da limitação de fluxo em interseções e usando dados sobre a estrutura de redes urbanas reais, simula a viagem de agentes em três tipos de cenários disruptivos. Ao final, sua análise foi aplicada em partes das redes de São Francisco, São Paulo e Paris; para avaliar não apenas impactos sobre os sistemas de transporte, mas também os impactos ocorridos nos padrões de viagem e entre os resultados obtidos. O autor alerta sobre como os sistemas com maior

conectividade podem ser mais afetados por danos em suas interseções semaforizadas, e que dependendo de um valor de elementos afetados, esses agentes ficam indiferentes entre alterar o trajeto para escapar das interseções afetadas ou usar o mesmo trajeto em condições normais.

Finalmente, uma semelhança é traçada entre a gestão de tráfego tradicional e o estudo proposto.

3 MÉTODO E TÉCNICAS DE PESQUISA

De acordo com (GIL, 2008), o delineamento de pesquisa é considerar o ambiente onde os dados são coletados e os meios de controle das variáveis. É a fase do estudo na qual se considera a aplicação de métodos discretos para utilizar-se na investigação; ou seja, diz respeito à maneira como o trabalho é realizado.

Segundo (MARCONI e LAKATOS, 2003), o procedimento adotado para a coleta de dados é dividido em dois grupos:

Fontes de papel: que incluem pesquisa bibliográfica e documental;

Informações obtidas através de pessoas ou experimentos: que engloba pesquisa experimental, ex-post facto, levantamento, estudo de caso, pesquisa ação e pesquisa participante.

Este trabalho emprega em sua maior parte a pesquisa bibliográfica para extrair informações sobre os algoritmos em visão computacional e áreas correlatas, aplicando técnicas de análise de conteúdo e coletando dados amplos sobre diferentes fenômenos. Em determinados momentos, se torna uma pesquisa documental, ao citar estatísticas e dados oficiais pontualmente.

Por tratar-se ainda de comparativos entre os algoritmos de visão computacional, aplicados ao cenário urbano, este trabalho é considerado como pesquisa experimental, pois foi avaliado o objeto de estudo e considerado as variáveis que influenciam nos resultados ao analisar o desempenho individual e utilizando-se o mesmo material/cenário para cada teste realizado com um algoritmo e/ou sistema de visão computacional diferente.

Este trabalho aborda o desempenho de diferentes algoritmos e sistemas na aplicação da visão computacional ao identificar e rastrear em meio ao tráfego urbano; detectando pedestres e/ou veículos em cenas gravadas capturando o tráfego.

A pesquisa é direcionada para o futuro desenvolvimento de protótipo a ser implementado na análise de caso e na sua viabilidade técnica. A metodologia utilizada visa investigar, analisar e realizar o estudo de referências, investigando a produção científica disponível sobre:

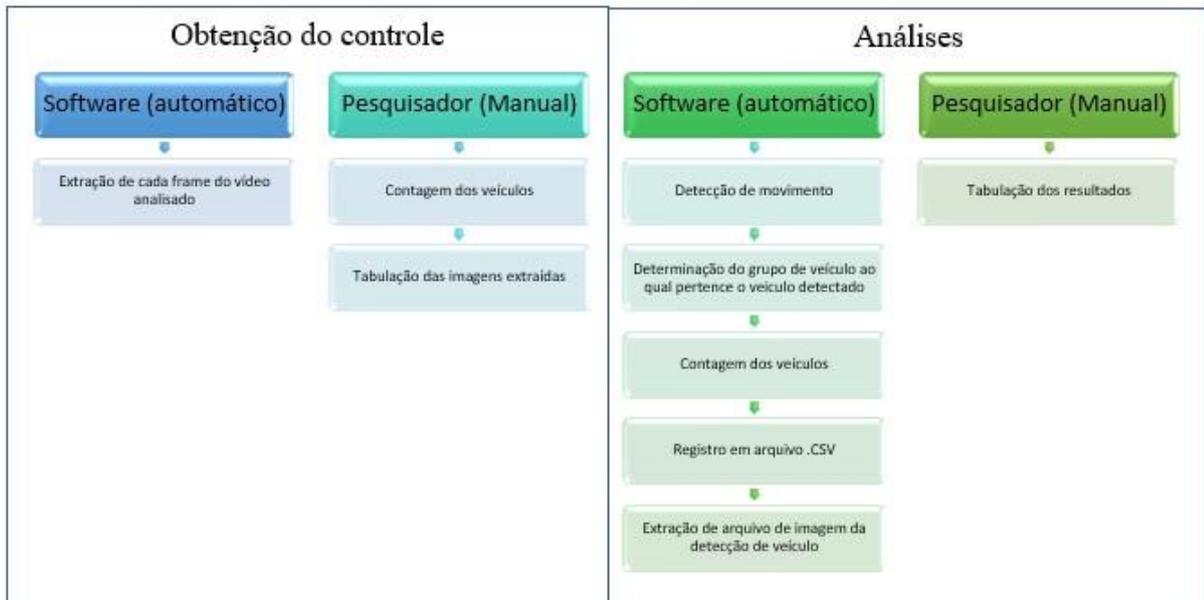
Semáforos inteligentes gerenciados por sistema de visão computacional;

Gestão de tráfego urbano em cidades inteligentes com base em visão computacional;

Sistemas de visão computacionais e de IA para uso em controle de tráfego e identificação de veículos e pessoas.

Estudos de casos aplicados, visando entender quais são as melhores práticas aplicadas, suas dificuldades, influências e impactos no cenário urbano e desafios tecnológicos;

Diagrama 14 - Procedimentos realizados para análise dos algoritmos



Fonte: O autor

3.1 Classificação da pesquisa

Este trabalho foi idealizado com natureza aplicada, buscando gerar conhecimentos para solucionar problemas na prática; de abordagem quantitativa; objetivos exploratórios e descritivos; com procedimentos bibliográficos, documental, porém com forte características experimentais, uma vez que os vídeos utilizados retratam diferentes situações de tráfego de veículos e pedestres; o que poderia também ser considerado como um estudo de laboratório por utilizar-se de vídeos e os eventos/fenômenos serem sempre os mesmos a cada amostragem de testes.

3.2 Descrição do método - literatura

Inicialmente idealizado seguindo os moldes de um Mapeamento Sistemático da Literatura, ou seja, desenvolvendo-se uma *string* de busca, no entanto, mesmo tentando melhorar essa *string* o número de artigos que retornava era incapaz de contemplar todas as possibilidades aplicáveis, seja pela quantidade de artigos retornados ou a quantidade de artigos não utilizáveis.

Como alternativa, foi proposto à realização de múltiplas buscas, tanto com base em *strings*, como de forma manual, direcionados a temas relevantes, porém distintos uns dos outros, mas com forte ligação com o assunto central desta pesquisa, visão computacional; portanto estes artigos, poderiam apresentar um conjunto de informações que integradas e observadas sobre perspectiva diferente seriam importantes para o entendimento do tema central.

O estudo da revisão de literatura científica foi realizado em artigos publicados no **período de 01/01/1970 a 31/12/2022** e coletados nas bases de dados citadas no Quadro 2.

Quadro 2 - Bases de dados consultadas

Base de dados	Endereço web
ACM	http://dl.acm.org
IEEE	http://ieeexplore.ieee.org/Xplore
RESEARCHGATE	https://www.researchgate.net/
UCLA	https://escholarship.org/uc/ucla
ARXIV	https://arxiv.org/
Web of Science	https://www.webofscience.com/
VISIONBIB	http://www.visionbib.com/
ACADEMIA	https://www.academia.edu/
IJCATR	https://ijcat.com/
SCIENCE DIRECT	https://www.sciencedirect.com/
Google Scholar	https://scholar.google.com.br/

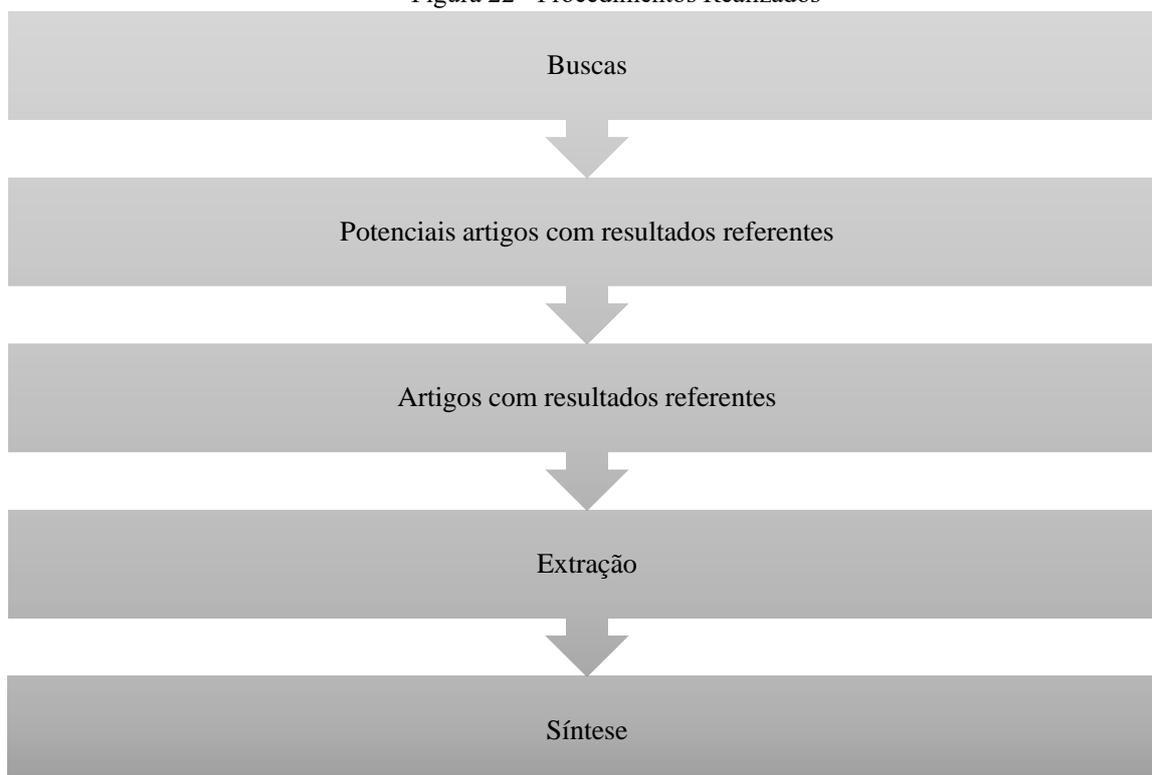
Fonte: O Autor

Os artigos primários deveriam relatar/estudar algoritmos em visão computacional, ou seja, algum tipo de algoritmo aplicado a visão computacional deveria ser mencionado nos artigos retornados. Nesta etapa, cada estudo foi lido e analisado de maneira independente, para determinar se eles deveriam passar ou não para próxima fase a partir de seus títulos, resumos e conclusões. Caso tivesse qualquer dúvida sobre a aceitação do artigo, ele era incluído para a próxima fase.

Dessa forma, os estudos que passaram pela etapa anterior foram separados em uma lista de estudos considerados potencialmente relevantes, onde todos os estudos pré-selecionados foram incluídos.

Na etapa seguinte, os estudos que eram potencialmente relevantes passaram então por uma nova seleção, mais rigorosa, que consistia na leitura completa dos artigos e na aplicação dos critérios de inclusão e exclusão, resultando assim em uma lista de artigos selecionados para esta pesquisa. A Figura 22 apresenta os passos dessa pesquisa e seus critérios serão explicados na sequência.

Figura 22 - Procedimentos Realizados



Fonte: O Autor

E se utilizou das combinações dos termos nas buscas, conforme demonstra os quadros de 03 ao 07, nos sites que não possuem sistema de strings de comando de query direta, nos que se aplicam os comandos aplicados e retornos de artigos, foram:

ACM - **"query": {Abstract:("Computer Vision" AND "Algorithms")} AND Keyword:("Computer Vision" OR "Algorithms" OR "Traffic control")}** **"filter": {Media Format: PDF},{Publication Date: (01/01/1970 TO 08/31/2022)},{ACM Content: DL}**

Retornando 93 artigos, sendo 1 duplicado (mesmo conteúdo, porém publicado em dois jornais diferentes); após uma seleção prévia, 11 artigos foram selecionados para uma avaliação mais aprofundada.

Quadro 3 - Combinações dos termos de busca | UCLA

Fonte:	UCLA	
String de Busca:	"computer vision" and "Algorithms"	
Filtros aplicados		
Type of Work:	X	Article (292)
	-	Book (1)
	-	Theses (0)
	-	Multimedia (1)
Peer Review:	X	Peer Review
Publication Year:	From: 1970 To: 2022	
Campus:	X	UC BERKELEY (22)
	X	UC DAVIS (114)
	X	UC IRVINE (43)
	X	UCLA (45)
	X	UC MERCED (2)
	X	UC RIVERSIDE (24)
	X	UC SAN DIEGO (24)
	-	UCSF (0)
	X	UC SANTA BARBARA (6)
	X	UC SANTA CRUZ (25)
	X	UC OFFICE OF THE PRESIDENT (4)
	-	LAWRENCE BERKELEY NATIONAL LABORATORY (0)
	-	UC AGRICULTURE & NATURAL RESOURCES (0)
Discipline:	X	Engineering (207)
	-	Social and Behavioral Sciences (134)
	X	Physical Sciences and Mathematics (93)
	-	Life Sciences (13)
	-	Medicine and Health Sciences (13)
	-	Arts and Humanities (6)
	-	Education (4)
	-	Architecture (2)
	-	Business (2)
	-	Law (2)
Observações.:		
a) Números entre parênteses indicam a quantidade de artigos retornados		
b) X = Selecionado		
RETORNOU 292 ARTIGOS		

Fonte: O Autor

Quadro 4 - Combinações dos termos de busca | ACM Digital Library

Fonte:	ACM	
String de Busca:	"query": {Abstract:"computer vision"} AND Abstract:"Algorithms"} "filter": {Article Type: Research Article},{E-Publication Date: (01/01/1970 TO 08/31/2022)},{ACM Content: DL}	
Filtros aplicados		
Publications / All Publications:	X	Proceedings (359)
	X	Journals (54)
	X	Newsletters (10)
	X	Magazines (4)
Content Type:	X	Research Article (424)
	-	Short Paper (0)
	-	Poster (0)
	-	Abstract (0)
	-	Demonstration (0)
	-	More (0)
Media Formats:	X	PDF (425)
	-	Image (89)
	-	HTML (81)
	-	Video (32)
	-	Archive/ZIP (17)
	-	More (3)
Observações.:		
a) Números entre parênteses indicam a quantidade de artigos retornados		
b) X = Selecionado		
RETORNOU 424 ARTIGOS		

Fonte: O Autor

Quadro 5 - Combinações dos termos de busca | ACM Digital Library - OpenCV

Fonte:	ACM	
String de Busca:	"query": {AllField:(opencv)} "filter": {ACM Pub type: Journals},{Article Type: Research Article},{ACM Content: DL}	
Filtros aplicados		
Publications / All Publications:	- X - - -	Proceedings (2301) Journals (304) Newsletters (34) Books (19) Magazines (10)
Content Type:	X - - - - -	Research Article (2668) Short Paper (176) Poster (172) Abstract (98) Demonstration (86) More (18)
Media Formats:	X - - - - -	PDF (3590) Image (873) HTML (559) Video (485) Archive/ZIP (153) More (4)
Observações.:		
a) Números entre parênteses indicam a quantidade de artigos retornados		
b) X = Selecionado		
RETORNOU 304 ARTIGOS		

Fonte: O Autor

Quadro 6 - Combinações dos termos de busca | ACM Digital Library - YOLO

Fonte:	ACM	
String de Busca:	"query": {AllField:(yolo)} "filter": {ACM Pub type: Journals},{Article Type: Research Article},{ACM Content: DL}	
Filtros aplicados		
Publications / All Publications:	-	Proceedings (1214)
	X	Journals (134)
	-	Magazines (6)
	-	Newsletters (5)
Content Type:	X	Research Article (1140)
	-	Short Paper (69)
	-	Poster (37)
	-	Survey (19)
	-	Demonstration (13)
	-	More (9)
Media Formats:	X	PDF (1357)
	-	Image (514)
	-	HTML (456)
	-	Video (113)
	-	Fulltext (28)
	-	More (3)
Observações.:		
a) Números entre parênteses indicam a quantidade de artigos retornados		
b) X = Selecionado		
RETORNOU 109 ARTIGOS		

Fonte: O Autor

Quadro 7 - Combinações dos termos de busca | ScienceDirect - BGS

Fonte:		ScienceDirect
Filtros aplicados		
Refine by:	X	Subscribed journals
Years:	X	2022 (38)
	X	2021 (24)
	X	2020 (12)
	X	2019 (7)
	X	2018 (22)
	X	2017 (19)
	X	2016 (17)
	X	2015 (17)
	X	2014 (13)
	X	2013 (32)
	X	2012 (23)
	X	2011 (11)
	X	2010 (8)
	X	2009 (4)
	X	2008 (4)
	X	2007 (3)
	X	2006 (4)
	X	2005 (6)
	X	2004 (11)
	X	2003 (3)
X	2002 (2)	
X	2001 (5)	
X	2000 (2)	
X	1999 (1)	
X	1998 (1)	
Article Type:	X	Research articles (356)
Publication title:	X	Applied Mathematical Modelling (85)
	X	Theoretical Computer Science (55)
	X	Computers & Mathematics with Applications (43)
	X	Mathematical and Computer Modelling (30)
	X	Science of Computer Programming (24)
	X	Knowledge-Based Systems (18)
	X	Journal of Computer and System Sciences (18)
	X	Artificial Intelligence (15)
	X	Applied Soft Computing (15)
	X	Applied Energy (14)
	X	Computers & Security (15)
	X	Pattern Recognition (15)
X	Engineering Applications of Artificial Intelligence (9)	
Subject areas:	X	Engineering (244)
	X	Computer Science (227)
Access Type:	X	Open access & Open archive (356)
Observações.:		
a) Números entre parênteses indicam a quantidade de artigos retornados		
b) X = Selecionado		
RETORNOU 356 ARTIGOS		

Fonte: O Autor

Após executadas as buscas, foi realizada a análise e leitura dos resumos dos artigos encontrados, conforme os seguintes critérios de inclusão: artigos publicados em inglês, português ou espanhol. Como critério de exclusão foi adotado: artigos que não tratavam exclusivamente de ferramentas de autoria.

Devido a quantidade de artigos e websites pesquisados, optou-se por utilizar de ferramentas que na visão do autor, considera-se automatizadas e inteligentes, para se estabelecer o controle e a dinâmica de um mapeamento sistemático de literatura, ao menos, no que tange ao trabalho das fases iniciais deste procedimento. Tendo em vista que não houve outros pesquisadores realizando a filtragem e seleção, não é considerável pelo pesquisador, como um “real” mapeamento sistemático de literatura; mas utilizou-se de suas reais atribuições para se estabelecer o material que compõe a revisão sistemática de literatura empregada nesse trabalho.

Para auxiliar no mapeamento e revisão, no que tange a organização e suporte as etapas de filtragens, foi escolhida a ferramenta de apoio: *StArt* (*State of the Art through Systematic Review*), por ser uma ferramenta gratuita e desenvolvida no LaPES (Laboratório de Pesquisa em Engenharia de Software) da UFSCAR (Universidade Federal de São Carlos) encontrando-se disponível em <http://lapes.dc.ufscar.br/ferramentas/start-tool> (link atualizado em 04/10/2022).

Utilizando-se como ferramenta o *software StArt* e os seguintes critérios de inclusão:

- Artigos completos publicados em inglês, português ou espanhol em revistas ou conferências que descrevam estudos sobre os temas analisados;
- Material disponível de forma integral e gratuita;

A partir da análise do título, palavras-chave, resumo e conclusão, foram excluídos os estudos que se enquadravam nos critérios de exclusão adotados:

- Artigos que não tratavam ou de baixa aderência ao tema principal (claramente irrelevantes para a pesquisa, levando em consideração as questões investigadas);
- Acesso ao material somente de forma paga ou similar;
- Artigos disponibilizados em outros idiomas;

A extração foi realizada de forma individual, com o preenchimento dos campos no *software StArt*. Durante esse processo, o pesquisador tentou extrair as relações de maneira clara e que os resultados estivessem contidos no texto de maneira que também ficasse claro para o leitor o que estava ocorrendo. O pesquisador podia rever, expandir ou limitar a definição das variáveis, mas sempre que isso ocorria, ele retornava aos relacionamentos extraídos

anteriormente para buscar novos relacionamentos que pudessem se enquadrar com a nova definição.

Ao final do processo do 2º filtro, as publicações selecionadas foram catalogadas utilizando-se na maioria das vezes, o *Mendeley*, que é gratuito e foi utilizado no gerenciamento dos artigos científicos e dos metadados. Existe a versão *Web* e a *Cliente*, ambas disponíveis em <http://www.mendeley.com> (link atualizado em 04/10/2022)

3.3 Descrição do método – procedimento dos testes

Utilizando-se de vídeo contendo cenários e em condições reais; para que pudesse ser realizadas as comparações de desempenho entre os algoritmos, estabeleceu-se alguns critérios em relação aos testes práticos:

Linguagem de programação: Python

Biblioteca de visão computacional: OpenCV

Hardware e softwares idênticos para todos os testes, vide o

APÊNDICE C – Equipamentos utilizados para mais detalhes;

O software de teste deveria realizar:

- Contar dois grupos de veículos: Carros e Caminhões, em quaisquer sentidos de direção (subindo ou descendo);
- Registrar através de arquivo cada grupo de veículo identificado;
- Nomear cada arquivo de imagem com: ID, Grupo e Sentido de direção;
- Registrar em arquivo .CSV: Data e horário do registro, Grupo e Sentido de direção;
- Registrar em arquivo de vídeo o processo dos testes, demonstrando o *ROI* e o contador de veículos.

No processo, o pesquisador criou um *software* com uma interface gráfica, para que o leitor consiga reproduzir o processo de testes ou estudar os algoritmos em questão e este encontra-se disponível em: <https://rsb.net.br/mestrado/>

Os algoritmos de subtração de imagem e rastreamento foram testados quanto a sua eficácia de forma geral, na contagem de veículos, divididos em dois grupos distintos apenas devido a área que ocupam em pixels, nesse contexto, adaptando o código fonte (disponibilizado como livre), contabilizando a quantidade de veículos e caminhões, foram realizados dez conjuntos de testes, onde se obteve de cada:

Um vídeo gravado com os resultados e o comportamento do algoritmo testado;

Imagens coletadas automaticamente da *ROI* para cada detecção há um arquivo de imagem respectivamente;

Um relatório .CSV com o registro de cada detecção

Após as coletas realizadas em datas e horários aleatórios, sem utilização de aprendizado de máquina e sem intervenções do pesquisador além da seleção manual da *ROI*, cada algoritmo testado teve seu comportamento registrado, e ao final foram estudados os registros obtidos.

Inicialmente, a primeira análise foram as contagens dos veículos detectados, onde foi possível observar os resultados na Tabela 4.

Para esta análise foi necessário o estabelecimento do registro de controle; onde, através de um software para extração de frames do vídeo, código disponível no APÊNDICE A – Código

fonte: Obtenção dos frames de vídeos e obtendo 25.189 arquivos de imagem; cada uma representa um frame do vídeo “Traffic_3.mp4”.

É possível visualizar o arquivo original através do link do Youtube: <https://youtu.be/6Yiuibs7MaY> e sua condição inicial é demonstrada na Imagem 6.

Este vídeo foi escolhido por ser o mais próximo dos aplicados ao benchmark CDNET, nele encontra-se veículos em diferentes configurações, velocidade, marca e modelo, além de cores, além de se encontrar veículos nos dois sentidos. Cenário muito similar aos encontrados em muitas cidades com grande fluxo de veículos. Por se tratar de um cenário desafiador para o sistema de visão computacional provar sua acurácia, e por estar muito relacionado aos veículos e não tanto aos pedestres, o pesquisador focou a maior parte dos testes práticos para este cenário; ou seja, direcionou aos veículos, porém a adequação ao uso para detecção de pedestres, é praticamente o mesmo conceito, pois todos os algoritmos detectam o movimento na imagem analisada, sendo desta forma, ignorado o tipo de objeto detectado, mas sim seu movimento dentro da cena analisada.

Imagem 6 – Quadro (frame) inicial do vídeo "Traffic_3.mp4"



Fonte: O autor com base no vídeo obtido de (GRANATYR, VARGAS e , 2022)

Com base nestes arquivos e de forma manual; foram contabilizados e registrados os veículos; resultando em 451 arquivos de imagens, selecionadas por registrarem 869 carros, 72 caminhões e 3 motocicletas.

Na Tabela 20 há o registro de cada veículo do grupo e em qual arquivo de imagem está registrado/contabilizado. Estes arquivos encontram-se no APÊNDICE E – Frames extraídos do vídeo Traffic_3.mp4 e se encontra disponível em <https://rsb.net.br/mestrado/>

Tabela 4 - Dados obtidos nos testes comparativos com o vídeo Traffic_3.mp4

Algoritmos	GMG	MOG	MOG2	KNN	CNT	LSBP	GSOC
Comparativo 01							
QTD total de carros	452	481	432	396	170	496	453
QTD total de caminhões	116	83	64	115	92	133	52
Comparativo 02							
QTD total de carros	437	379	470	516	120	484	470
QTD total de caminhões	115	52	45	101	74	125	81
Comparativo 03							
QTD total de carros	441	420	433	532	120	514	454
QTD total de caminhões	111	57	42	120	71	119	78
Comparativo 04							
QTD total de carros	455	443	458	508	123	493	465
QTD total de caminhões	112	49	49	123	74	126	83
Comparativo 05							
QTD total de carros	445	441	430	521	117	441	477
QTD total de caminhões	114	57	45	100	79	123	84
Comparativo 06							
QTD total de carros	483	435	481	547	120	511	470
QTD total de caminhões	117	51	54	111	76	121	80
Comparativo 07							
QTD total de carros	451	455	441	524	139	500	416
QTD total de caminhões	110	52	35	111	77	128	76
Comparativo 08							
QTD total de carros	462	439	433	525	119	500	446
QTD total de caminhões	116	50	37	117	84	123	77

Fonte: O autor

Foram realizados oito grupos de comparativos com o mesmo vídeo, sendo analisado o comportamento dos diferentes algoritmos e o mesmo algoritmo foi testado (oito vezes) em dias e horários diferentes. Todos tiveram seus dados registrados e disponibilizados para análise em:

<https://rsb.net.br/mestrado/>

O pesquisador não alterou os parâmetros dos filtros entre os testes, ou seja, a única alteração entre cada teste, foram os diretórios onde seriam salvos os dados obtidos.

Assim, após uma série de testes prévios para definir os melhores valores para os filtros dos algoritmos, os testes apresentados neste trabalho foram realizados com os valores fixados para os seguintes algoritmos: GMG, MOG, MOG2, KNN e CNT.

A seguir, detalhamentos dos parâmetros dos filtros aplicados nos algoritmos citados, segundo a documentação do OpenCV em: <https://docs.opencv.org/>

Quadro 8 – Filtros aplicados nos algoritmos: GMG, MOG, MOG2, kNN e CNT

```

if BGS_TYPE == "GMG":
    return
cv2.bgsegm.createBackgroundSubtractorGMG(initializationFrames=120,
decisionThreshold=.8)

if BGS_TYPE == "MOG":
    return cv2.bgsegm.createBackgroundSubtractorMOG(history=200,
nmixtures=5, backgroundRatio=.7, noiseSigma=0)

if BGS_TYPE == "MOG2":
    return cv2.createBackgroundSubtractorMOG2(history=50,
detectShadows=False, varThreshold=200)

if BGS_TYPE == "KNN":
    return cv2.createBackgroundSubtractorKNN(history=100,
dist2Threshold=400, detectShadows=True)

if BGS_TYPE == "CNT":
    return cv2.bgsegm.createBackgroundSubtractorCNT(minPixelStability=15,
useHistory=True, maxPixelStability=15 * 60, isParallel=True)

```

Fonte: Adaptação realizada no código fonte fornecido por (GRANATYR, VARGAS e , 2022)

Para GMG, `initializationFrames=120`, são utilizados os primeiros 120 quadros do vídeo para formar o background (BG), `decisionThreshold=8`, retorna o valor do limite de decisão que está fixado em 8, esse valor de decisão é o valor acima do qual o pixel é determinado como foreground (FG). (OPENCV.ORG (7), 2022)

Temos para os algoritmos MOG, MOG2 e KNN, há o campo `history`; este é o valor da quantidade de frames que são analisados para formação da subtração do fundo, ou seja, duração da história. (OPENCV.ORG, 2022)

Para MOG, `nmixtures` é o parâmetro de números de misturas gaussianas a serem empregado, `backgroundRatio` é a proporção de plano de fundo e `noiseSigma` é a intensidade do ruído na imagem, ou seja, desvio padrão do brilho ou de cada canal de cor, onde o valor zero significa valor automático. (OPENCV.ORG (5), 2022)

Para MOG2, `detectShadows` é a detecção de sombras, se verdadeiro, o algoritmo detectará sombras e as marcará, mas ele diminui a velocidade, portanto, foi considerado como não necessário durante os testes e foi definido como falso; enquanto o `varThreshold`, é o valor do limiar na distância de Mahalanobis ao quadrado entre o pixel e o modelo para decidir se um pixel é bem descrito pelo modelo de fundo, ou seja, este parâmetro não afeta a atualização em segundo plano. (OPENCV.ORG (6), 2022)

Para kNN, `dist2Threshold` é o limiar na distância ao quadrado entre o pixel e a amostra para decidir se um pixel está próximo dessa amostra, como não afeta a atualização em segundo plano, foi experimentado valores diferentes e mantido o que o pesquisador identificou em sua opinião como um intermediário entre os valores aleatórios testados anteriormente; enquanto `detectShadows` é similar ao parâmetro do MOG2. (OPENCV.ORG (5), 2022).

Em CNT; `minPixelStability` é o número de quadros com a mesma cor de pixel para considerar estável; `useHistory` determina se é dado crédito a um pixel por ser estável por muito tempo; enquanto `maxPixelStability` é o crédito máximo permitido para um pixel no histórico e `isParallel` determina se estamos paralelizando o algoritmo. (OPENCV.ORG (8), 2022)

Os algoritmos LSBP e GSOC não tiveram filtros inseridos ou alteração dos valores padronizados, sendo assim, foram testados nas condições apresentadas a seguir:

Quadro 9 – Detalhe da chamada sem filtros nos algoritmos: LSBP e GSOC

```

if BGS_TYPE == "LSBP":
    return cv2.bgsegm.createBackgroundSubtractorLSBP()
if BGS_TYPE == "GSOC":
    return cv2.bgsegm.createBackgroundSubtractorGSOC()

```

Fonte: Adaptação realizada no código fonte fornecido por (GRANATYR, VARGAS e , 2022)

O código fonte completo encontra-se no APÊNDICE B – Código fonte: Testes de desempenho dos algoritmos BGS e no GitHub:

<https://github.com/RogérioSantinoBarboza/StartHere>

3.3.1 Análises estatísticas

Para a realização das análises foram utilizadas as ferramentas: *JAMOVI* e *JASP*.

JAMOVI, que se trata de um *software* estatístico *open source* com interface gráfica, do tipo modular, de uso gratuito e disponível em <https://www.jamovi.org/download.html> (link atualizado em 04/01/2023) e a versão utilizada foi a 2.3.21.0; embora haja a versão em *Cloud*, foi utilizada a versão *Desktop*.

JASP, é um programa gratuito e de código aberto para análise estatística apoiado pela Universidade de Amsterdã, que foi projetado para ser fácil de usar e familiar aos usuários do SPSS, ainda oferece procedimentos de análise padrão em sua forma clássica e bayesiana. Disponível em <https://jasp-stats.org/> (link atualizado em 04/01/2023) e a versão utilizada foi a 0.16.4 para desktop.

3.3.1.1 Teste de normalidade (Shapiro-Wilk)

A hipótese nula do teste de Shapiro-Wilk é que a população possui distribuição normal e, portanto, um valor de $p < 0.05$ indica que se rejeite a hipótese nula, ou seja, que os dados **não** possuem distribuição normal. Para determinar isso, foi realizado uma análise estatística para se determinar se os dados obedeciam ou não aos parâmetros e utilizando-se de **estatística descritiva**, obtendo-se os dados apresentados na Tabela 5 (p. 131).

Tabela 5 – Análise descritiva para determinar a normalidade dos dados obtidos

	Algoritmo	Mediana	Desvio-padrão	Variância	Mínimo	Máximo	Assimetria		Shapiro-Wilk		Percentis		
							Assimetria	Erro-padrão	W	p	25th	50th	75th
Carros	GMG	451.5	14.41	207.64	437	483	1.292	0.752	0.906	0.325	444.0	451.5	456.8
	MOG	440.0	29.21	853.13	379	481	-0.780	0.752	0.924	0.467	431.3	440.0	446.0
	MOG2	437.0	19.81	392.50	430	481	0.900	0.752	0.828	0.057	432.8	437.0	461.0
	kNN	522.5	46.93	2202.27	396	547	-2.486	0.752	0.658	<.001	514.0	522.5	526.8
	CNT	120.0	18.13	328.86	117	170	2.180	0.752	0.658	<.001	119.8	120.0	127.0
	LSBP	498.0	22.85	521.98	441	514	-1.896	0.752	0.803	0.031	490.8	498.0	502.8
	GSOC	459.5	19.37	375.13	416	477	-1.377	0.752	0.881	0.193	451.3	459.5	470.0
Caminhões	GMG	114.5	2.59	6.70	110	117	-0.411	0.752	0.922	0.446	111.8	114.5	116.0
	MOG	52.0	11.16	124.55	49	83	2.452	0.752	0.652	<.001	50.8	52.0	57.0
	MOG2	45.0	9.38	87.98	35	64	0.820	0.752	0.947	0.681	40.8	45.0	50.3
	kNN	113.0	8.33	69.36	100	123	-0.472	0.752	0.926	0.480	108.5	113.0	117.8
	CNT	76.5	6.74	45.41	71	92	1.321	0.752	0.888	0.224	74.0	76.5	80.3
	LSBP	124.0	4.37	19.07	119	133	0.787	0.752	0.960	0.814	122.5	124.0	126.5
	GSOC	79.0	10.24	104.84	52	84	-2.413	0.752	0.683	0.001	76.8	79.0	81.5

Fonte: O Autor

Tabela 6 - Comparativo da taxa de acerto dos algoritmos em relação ao controle

	CARROS							CAMINHÕES						
	GMG	MOG	MOG2	KNN	CNT	LSBP	GSOC	GMG	MOG	MOG2	KNN	CNT	LSBP	GSOC
Comparativo 01	-47,99%	-44,65%	-50,29%	-54,43%	-80,44%	-42,92%	-47,87%	61,11%	15,28%	-11,11%	59,72%	27,78%	84,72%	-27,78%
Comparativo 02	-49,71%	-56,39%	-45,91%	-40,62%	-86,19%	-44,30%	-45,91%	59,72%	-27,78%	-37,50%	40,28%	2,78%	73,61%	12,50%
Comparativo 03	-49,25%	-51,67%	-50,17%	-38,78%	-86,19%	-40,85%	-47,76%	54,17%	-20,83%	-41,67%	66,67%	-1,39%	65,28%	8,33%
Comparativo 04	-47,64%	-49,02%	-47,30%	-41,54%	-85,85%	-43,27%	-46,49%	58,33%	-20,83%	-37,50%	38,89%	9,72%	70,83%	16,67%
Comparativo 05	-48,79%	-49,25%	-50,52%	-40,05%	-86,54%	-49,25%	-45,11%	58,33%	-20,83%	-37,50%	38,89%	9,72%	70,83%	16,67%
Comparativo 06	-44,42%	-49,94%	-44,65%	-37,05%	-86,19%	-41,20%	-45,91%	62,50%	-29,17%	-25,00%	54,17%	5,56%	68,06%	11,11%
Comparativo 07	-48,10%	-47,64%	-49,25%	-39,70%	-84,00%	-42,46%	-52,13%	52,78%	-27,78%	-51,39%	54,17%	6,94%	77,78%	5,56%
Comparativo 08	-46,84%	-49,48%	-50,17%	-39,59%	-86,31%	-42,46%	-48,68%	61,11%	-30,56%	-48,61%	62,50%	16,67%	70,83%	6,94%

Fonte: O Autor

3.3.1.2 Controle manual dos dados

O controle, realizado manualmente, contando todos os veículos contidos no vídeo de origem, utilizando-se do programa de subtração de frames, obtendo assim condições para a contagem, pois cada frame que forma o vídeo, transformou-se em um arquivo de imagem e os resultados tabulados como apresentado na Tabela 7.

Tabela 7 - Controle manual da contagem de veículos

QTD de carros	869
QTD de caminhões	72

Fonte: O Autor

Note que o valor p para os algoritmos de BGS testados, alternam, sendo que para o grupo de contagem de CARROS, obteve os valores apresentados da Tabela 8 e para o grupo de contagem de CAMINHÕES, obteve os valores da Tabela 9.

Tabela 8 - Observações dos resultados de valores p (Shapiro-Wilk) | CARROS

Algoritmo	Valor p	Observações
GMG	0.325	
MOG	0.467	Valor maior que 0.05 , portanto os dados possuem distribuição normal
MOG2	0.057	
GSOC	0.193	
KNN	< 0.001	Valor menor que 0.05 , portanto os dados não possuem distribuição normal
CNT	< 0.001	
LSBP	0.031	

Fonte: O Autor

Tabela 9 - Observações dos resultados de valores p (Shapiro-Wilk) | CAMINHÕES

Algoritmo	Valor p	Observações
GMG	0.446	
MOG2	0.681	
KNN	0.480	Valor maior que 0.05 , portanto os dados possuem distribuição normal
CNT	0.224	
LSBP	0.814	
MOG	< 0.001	Valor menor que 0.05 , portanto os dados não possuem distribuição normal
GSOC	0.001	

Fonte: O Autor

3.3.1.3 Análise estatística da amostragem de precisão dos algoritmos

O controle foi realizado manualmente com base na análise dos frames extraídos do vídeo original, comparando com os dados obtidos pelo programa de testes, conforme descrito no item 4 e os resultados tabulados como apresentado nos itens 4 ao 4.7.

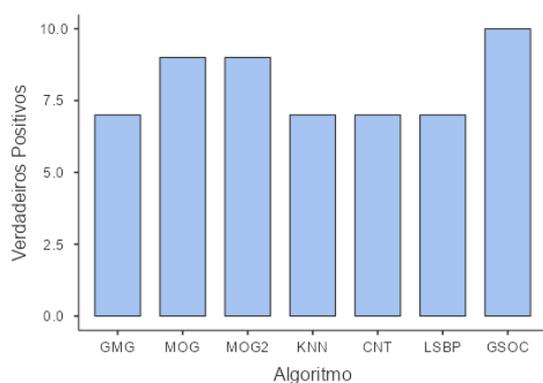
Tabela 10 - Estatística Descritiva - Precisão dos algoritmos da amostragem

	Algoritmo	Mediana
Verdadeiros Positivos	GMG	7
	MOG	9
	MOG2	9
	KNN	7
	CNT	7
	LSBP	7
	GSOC	10
Falsos Positivos	GMG	3
	MOG	1
	MOG2	1
	KNN	3
	CNT	3
	LSBP	3
	GSOC	0

Fonte: O Autor

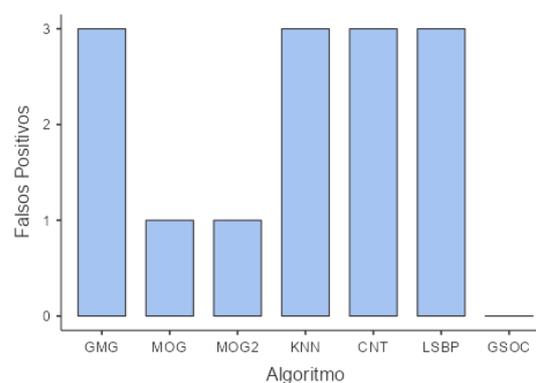
Na Tabela 10 encontra-se os valores referentes à acuraria dos algoritmos, onde entre os verdadeiros positivos o GSOC apresentou o melhor resultado, assim como em falsos positivos, seguido pelos algoritmos MOG e MOG2; como demonstrado nos Gráfico 5 e Gráfico 6.

Gráfico 5 – Verdadeiros Positivos



Fonte: O Autor

Gráfico 6 - Falsos Positivos



Fonte: O Autor

Tabela 11 - Teste t para uma amostra

		Estatística	gl	p
Verdadeiros Positivos	t de Student	-4.10	6.00	0.006

Nota. $H_a \mu \neq 10$

Fonte: O autor

Tabela 12 - Teste à Normalidade (Shapiro-Wilk) | Amostragem

	W	p
Verdadeiros Positivos	0.760	0.016

Nota. Um p -value pequeno sugere a violação do pressuposto da normalidade

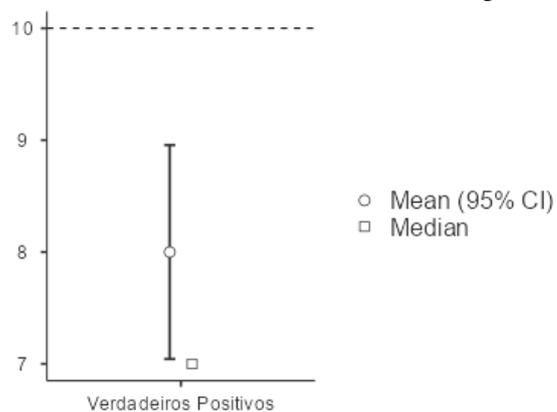
Fonte: O autor

Tabela 13 - Estatística Descritiva da Amostragem de verdadeiros positivos

	N	Média	Mediana	Desvio-padrão	Erro-padrão
Verdadeiros Positivos	7	8.00	7	1.29	0.488

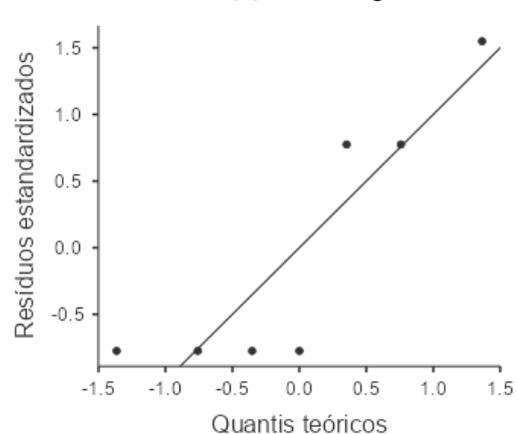
Fonte: O autor

Gráfico 7 - Verdadeiros Positivos - Amostragem



Fonte: O autor

Gráfico 8 - QQ - Amostragem



Fonte: O autor

Tabela 14 - Teste t para uma amostra - Falsos Positivos

		Estatística	gl	p
Falsos Positivos	t de Student	4.10	6.00	0.006

Nota. $H_a \mu \neq 0$

Fonte: O Autor

Tabela 15 - Teste à Normalidade (Shapiro-Wilk) - Amostragem - Falsos Positivos

	W	p
Falsos Positivos	0.760	0.016

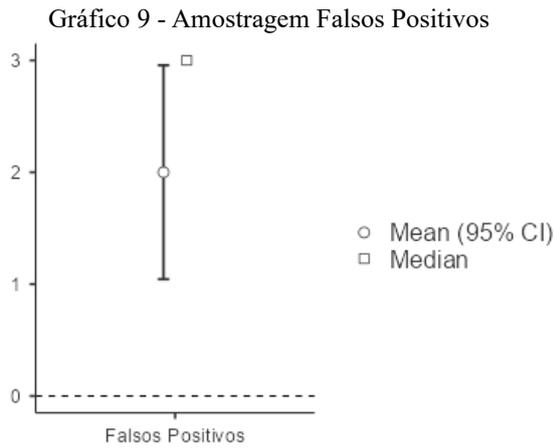
Nota. Um **p-value** pequeno sugere a violação do pressuposto da normalidade

Fonte: O Autor

Tabela 16 - Estatística Descritiva da amostragem de falsos positivos

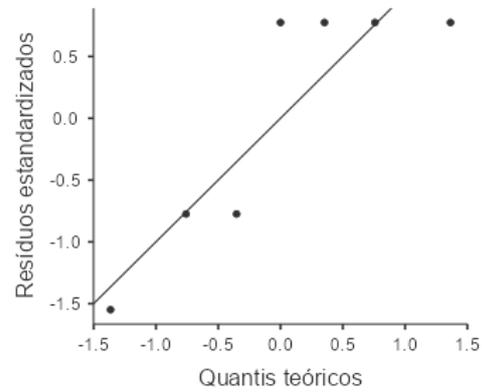
	N	Média	Mediana	Desvio-padrão	Erro-padrão
Falsos Positivos	7	2.00	3	1.29	0.488

Fonte: O Autor



Fonte: O Autor

Gráfico 10 - QQ da amostragem de falsos positivos



Fonte: O Autor

Tabela 17 - ANOVA a um fator (não-paramétrica) Kruskal-Wallis - Amostragem

	χ^2	gl	p
Verdadeiros Positivos	6.00	6	0.423
Falsos Positivos	6.00	6	0.423

Fonte: O Autor

Tabela 18 - Comparações múltiplas Dwass-Steel-Critchlow-Fligner - Verdadeiros Positivos

		W	p
GMG	MOG	1.41	0.954
GMG	MOG2	1.41	0.954
GMG	KNN	NaN	NaN
GMG	CNT	NaN	NaN
GMG	LSBP	NaN	NaN
GMG	GSOC	1.41	0.954
MOG	MOG2	NaN	NaN
MOG	KNN	-1.41	0.954
MOG	CNT	-1.41	0.954
MOG	LSBP	-1.41	0.954
MOG	GSOC	1.41	0.954
MOG2	KNN	-1.41	0.954
MOG2	CNT	-1.41	0.954
MOG2	LSBP	-1.41	0.954
MOG2	GSOC	1.41	0.954
KNN	CNT	NaN	NaN
KNN	LSBP	NaN	NaN
KNN	GSOC	1.41	0.954
CNT	LSBP	NaN	NaN
CNT	GSOC	1.41	0.954
LSBP	GSOC	1.41	0.954

Fonte: O Autor

Tabela 19 - Comparações múltiplas Dwass-Steel-Critchlow-Fligner - Falsos Positivos

		W	p
GMG	MOG	-1.41	0.954
GMG	MOG2	-1.41	0.954
GMG	KNN	NaN	NaN
GMG	CNT	NaN	NaN
GMG	LSBP	NaN	NaN
GMG	GSOC	-1.41	0.954
MOG	MOG2	NaN	NaN
MOG	KNN	1.41	0.954
MOG	CNT	1.41	0.954
MOG	LSBP	1.41	0.954
MOG	GSOC	-1.41	0.954
MOG2	KNN	1.41	0.954
MOG2	CNT	1.41	0.954
MOG2	LSBP	1.41	0.954
MOG2	GSOC	-1.41	0.954
KNN	CNT	NaN	NaN
KNN	LSBP	NaN	NaN
KNN	GSOC	-1.41	0.954
CNT	LSBP	NaN	NaN
CNT	GSOC	-1.41	0.954
LSBP	GSOC	-1.41	0.954

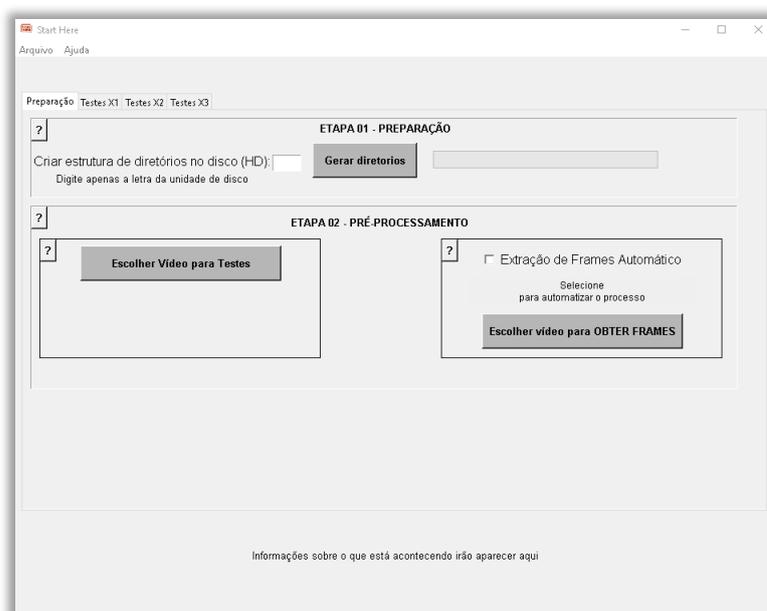
Fonte: O Autor

3.4 Descrição do software criado para replicação dos testes

Ao processar de forma manual, com as devidas alterações em determinadas linhas de programação, o pesquisador identificou uma possível dificuldade em replicar os testes aqui apresentados, ou até mesmo, pela dificuldade em estudar ou conhecer os algoritmos de visão computacional e sua aplicação prática, por leitores não habituados com linhas de programação em *PYTHON* ou conhecimentos em programação. Desta forma, decidiu-se por criar uma interface, que pode ser vista na sua versão 0.1 (Figura 23) e adicionar funcionalidades para replicar com maior fidelidade os processos realizados nesse trabalho. Isso será mantido de forma gratuita e disponibilizado através dos links abaixo, para futuros trabalhos e “refaturamento” do software.

<https://rsb.net.br/mestrado/>

Figura 23 - Interface criada para auxiliar o leitor



Fonte: O autor

4 ANÁLISE E INTERPRETAÇÃO DOS RESULTADOS

Este trabalho procurou abranger trabalhos desde 1956, com relatando como a “*Dartmouth Summer Research Project on Artificial Intelligence*”, contribuiu com o surgimento das experiências em visão computacional, com o uso das primeiras redes neurais para a detecção e classificação de objetos.

Como o período desta pesquisa abrange de 1950 até início de 2023, e por esse fato, seja considerado muito abrangente no contexto de tempo para uma tecnologia, o pesquisador considera relevante no contexto histórico, por ser uma área da ciência que depende de conhecimentos aprofundados de outras áreas do conhecimento como relatado e exemplificado na Figura 7; o que com avanço em outras áreas como relatado, acreditou-se que houve um avanço consequentemente em visão computacional.

Fato que é confirmado por algumas das referências, que afirmam que na década de 1990 devido ao avanço da internet, houve dados suficientes para que se disseminasse o reconhecimento de pessoas em vídeos e imagens digitais.

A definição de visão computacional, é um subgrupo da área de IA e DL; além de treinar as CNNs para desenvolver algo bem próximo do que seja a visão biológica para soluções comerciais, através de treinamentos específicos de CNNs para segmentação, classificação e detecção; com uso de imagens e vídeos como fontes de dados; conforme ilustrado no Quadro 1.

Enquanto para outras referências, os sensores, camera ou vídeos sejam os componentes básicos; descrevendo como a radiação da luz e espectro das cores é capturada e convertida em imagens em sistemas digitais, descrevendo os conceitos básicos e metodologias de processamento, codificação das imagens, segmentação, descrições, reconhecimento e interpretação das imagens digitais.

A introdução do rastreamento baseado em modelo de geometria como técnica de comparação para detectar e identificar objetos; utilizando de detecção de características do tipo Haar para detectar veículos com muita precisão e em tempo real, representando um grande avanço em relação ao trabalho dos autores anteriores, usando como um classificador do algoritmo *AdaBoost*.

O uso de um detector semelhante ao *Haar* para detecção dos veículos; funcionando com intensidades de pixel na janela de detecção, onde as diferenças dessas somas são calculadas e ajuda a classificar subseções da imagem.

A introdução do rastreamento por região de interesse, usando dessa técnica, conseguiram reduzir a complexidade de cálculos. A utilização de regiões de interesse para extraírem informações de cor e textura na imagem, utilizando PDI para segmentá-la com base na similaridade entre os pixels, seguindo o critério definido como: binarização, crescimento de regiões (*Region Growing*), divisão e junção das regiões (*Split & Merge*), aglomeração (*Clustering*) e limiares (*Thresholding*).

Outra abordagem tratada apresentou uma vantagem de encontrar alguns recursos diferenciáveis na imagem pelo qual o sistema pode detectar o objeto com mais eficiência, mesmo quando ocorre uma oclusão.

A subtração de fundo é uma técnica usada por muitas das referências, para segmentar a imagem e como abordaram em seus trabalhos, subtraindo pixel a pixel de uma imagem de fundo usada como referência. Também realizaram a separação dos objetos que se movimentam da parte estática da imagem.

Como é considerada pela maioria das referências como um problema para detecção de alterações em vídeos, que costuma ser a primeira etapa em soluções em visão computacional; que por este motivo, é necessário considerar possíveis fatores desafiadores, como: iluminação, oscilações, movimentos de objetos e entre outros.

O surgimento do CDnet2012 e de seu complemento em 2014, serve de benchmark para avaliar de forma independente e neutra os diversos algoritmos, independentemente de sua origem.

4.1 Controle de amostragem de precisão do algoritmo GMG

O controle foi realizado manualmente com base na análise dos frames extraídos do vídeo original, comparando com os dados obtidos pelo programa de testes; neste caso, foram um arquivo .CSV com o registro de data/hora, grupo de veículo com sentido de direção, e os resultados da amostra dos cinco primeiros registro de cada grupo, a confirmar: Carros e

Caminhões, estão tabulados como apresentado no Quadro 10 e as imagens obtidas estão no Quadro 22.

O material bruto encontra-se disponível em: <https://rsb.net.br/mestrado/>

Quadro 10 – Controle - Teste 01 - GMG

Item	Tempo,Veiculo	Arquivo	Avaliação
1	Sat Oct 22 23:17:33 2022,Carro Descendo 1	Carro_Descendo_0141.png	Verdadeiro carro
2	Sat Oct 22 23:17:35 2022,Carro Descendo 2	Carro_Descendo_0171.png	Verdadeiro carro
3	Sat Oct 22 23:17:39 2022,Carro Descendo 3	Carro_Descendo_0259.png	Verdadeiro carro
4	Sat Oct 22 23:17:41 2022,Carro Subindo 4	Carro_Subindo_0295.png	Verdadeiro carro
5	Sat Oct 22 23:17:42 2022,Caminhao Subindo 1	Caminhao_Subindo_0317.png	Falso caminhão
6	Sat Oct 22 23:17:42 2022,Carro Descendo 5	Carro_Descendo_0321.png	Verdadeiro carro
7	Sat Oct 22 23:17:48 2022,Caminhao Subindo 2	Caminhao_Subindo_0434.png	Verdadeiro caminhão
8	Sat Oct 22 23:17:54 2022,Caminhao Subindo 3	Caminhao_Subindo_0550.png	Verdadeiro caminhão
9	Sat Oct 22 23:18:13 2022,Caminhao Descendo 4	Caminhao_Descendo_0916.png	Falso caminhão
10	Sat Oct 22 23:18:21 2022,Caminhao Descendo 5	Caminhao_Descendo_1074.png	Falso caminhão

Fonte: O autor.

Os itens 4,5, 8, 9 e 10 do Quadro 10, apresentaram Oclusão, sendo considerados como falha do algoritmo.

4.2 Controle de amostragem de precisão do algoritmo MOG

O controle foi realizado manualmente com base na análise dos frames extraídos do vídeo original, comparando com os dados obtidos pelo programa de testes, conforme descrito no item anterior e os resultados tabulados como apresentado no Quadro 11 e as imagens coletadas estão no

Quadro 23.

Quadro 11 - Controle - Teste 01 - MOG

Item	Tempo,Veiculo	Arquivo	Avaliação
1	Sun Oct 23 00:01:08 2022,Carro Subindo 1	Carro_Subindo_0047.png	Verdadeiro carro
2	Sun Oct 23 00:01:09 2022,Carro Descendo 2	Carro_Descendo_0065.png	Verdadeiro carro
3	Sun Oct 23 00:01:10 2022,Carro Descendo 3	Carro_Descendo_0097.png	Verdadeiro carro
4	Sun Oct 23 00:01:12 2022,Carro Descendo 4	Carro_Descendo_0142.png	Verdadeiro carro
5	Sun Oct 23 00:01:13 2022,Carro Descendo 6	Carro_Descendo_0173.png	Verdadeiro carro
6	Sun Oct 23 00:01:19 2022,Caminhao Subindo 1	Caminhao_Subindo_0315.png	Falso caminhão
7	Sun Oct 23 00:01:23 2022,Caminhao Subindo 2	Caminhao_Subindo_0426.png	Verdadeiro caminhão
8	Sun Oct 23 00:01:28 2022,Caminhao Subindo 3	Caminhao_Subindo_0548.png	Verdadeiro caminhão
9	Sun Oct 23 00:01:51 2022,Caminhao Subindo 4	Caminhao_Subindo_1204.png	Verdadeiro caminhão
10	Sun Oct 23 00:01:57 2022,Caminhao Descendo 5	Caminhao_Descendo_1352.png	Verdadeiro caminhão

Fonte: O autor.

O item 6 do Quadro 11, apresenta Oclusão, sendo considerados como falha do algoritmo.

4.3 Controle de amostragem de precisão do algoritmo MOG2

O controle foi realizado manualmente com base na análise dos frames extraídos do vídeo original, comparando com os dados obtidos pelo programa de testes e os resultados tabulados como apresentado no Quadro 12 e as imagens coletadas estão no

Quadro 24.

Quadro 12 - Controle - Teste 01 – MOG2

Item	Tempo,Veiculo	Arquivo	Avaliação
1	Sun Oct 23 00:24:42 2022,Carro Descendo 1	Carro_Descendo_0095.png	verdadeiro carro

2	Sun Oct 23 00:24:43 2022,Carro Descendo 2	Carro_Descendo_0140.png	verdadeiro carro
3	Sun Oct 23 00:24:45 2022,Carro Descendo 3	Carro_Descendo_0172.png	verdadeiro carro
4	Sun Oct 23 00:24:48 2022,Carro Descendo 4	Carro_Descendo_0258.png	verdadeiro carro
5	Sun Oct 23 00:24:50 2022,Carro Subindo 5	Carro_Subindo_0296.png	verdadeiro carro
6	Sun Oct 23 00:24:53 2022,Caminhao Descendo 1	Caminhao_Descendo_0374.png	Falso caminhão
7	Sun Oct 23 00:25:31 2022,Caminhao Descendo 2	Caminhao_Descendo_1349.png	Verdadeiro caminhão
8	Sun Oct 23 00:25:31 2022,Caminhao Descendo 3	Caminhao_Descendo_1351.png	Verdadeiro caminhão
9	Sun Oct 23 00:26:00 2022,Caminhao Subindo 4	Caminhao_Subindo_2113.png	Verdadeiro caminhão
10	Sun Oct 23 00:26:10 2022,Caminhao Descendo 5	Caminhao_Descendo_2393.png	Verdadeiro caminhão

Fonte: O autor.

O item 6 do Quadro 12, apresenta Oclusão, sendo considerado como falha do algoritmo.

4.4 Controle de amostragem de precisão do algoritmo KNN

O controle foi realizado manualmente com base na análise dos frames extraídos do vídeo original, comparando com os dados obtidos pelo programa de testes, conforme descrito no item 4 e os resultados tabulados como apresentado no Quadro 13 e as imagens coletadas estão no

Quadro 25.

Quadro 13 - Controle - Teste 01 - KNN

Item	Tempo,Veiculo	Arquivo	Avaliação
1	Sat Oct 22 22:39:25 2022,Carro Descendo 1	Carro_Descendo_0049.png	Verdadeiro carro
2	Sat Oct 22 22:39:27 2022,Carro Descendo 2	Carro_Descendo_0088.png	Verdadeiro carro
3	Sat Oct 22 22:39:29 2022,Carro Descendo 4	Carro_Descendo_0132.png	Verdadeiro carro
4	Sat Oct 22 22:39:29 2022,Carro Subindo 3	Carro_Subindo_0126.png	Verdadeiro carro
5	Sat Oct 22 22:39:34 2022,Caminhao Descendo 1	Caminhao_Descendo_0251.png	Falso caminhão
6	Sat Oct 22 22:39:34 2022,Carro Subindo 5	Carro_Subindo_0246.png	Verdadeiro carro

7	Sat Oct 22 22:39:35 2022,Caminhao Descendo 2	Caminhao_Descendo_0287.png	Falso caminhão
8	Sat Oct 22 22:39:37 2022,Carro Descendo 6	Carro_Descendo_0313.png	Verdadeiro carro
9	Sat Oct 22 22:39:38 2022,Caminhao Subindo 3	Caminhao_Subindo_0342.png	Falso caminhão
10	Sat Oct 22 22:39:43 2022,Caminhao Subindo 4	Caminhao_Subindo_0448.png	Verdadeiro caminhão

Fonte: O autor.

Os itens 5 e 6 do Quadro 13, apresentaram Oclusão, e o item 9, apresenta falsa detecção, sendo considerados como falha do algoritmo.

4.5 Controle de amostragem de precisão do algoritmo CNT

O controle foi realizado manualmente com base na análise dos frames extraídos do vídeo original, comparando com os dados obtidos pelo programa de testes, conforme descrito no item 4 e os resultados tabulados como apresentado no Quadro 14 e as imagens coletadas estão listadas no

Quadro 26.

Quadro 14 - Controle - Teste 01 - CNT

Item	Tempo,Veiculo	Arquivo	Avaliação
1	Sun Oct 23 00:46:08 2022,Carro Subindo 1	Carro_Subindo_0045.png	verdadeiro carro
2	Sun Oct 23 00:46:09 2022,Carro Descendo 2	Carro_Descendo_0065.png	verdadeiro carro
3	Sun Oct 23 00:46:10 2022,Carro Descendo 3	Carro_Descendo_0096.png	verdadeiro carro
4	Sun Oct 23 00:46:10 2022,Carro Subindo 4	Carro_Subindo_0109.png	verdadeiro carro
5	Sun Oct 23 00:46:11 2022,Carro Descendo 5	Carro_Descendo_0141.png	verdadeiro carro
6	Sun Oct 23 00:46:16 2022,Caminhao Subindo 1	Caminhao_Subindo_0316.png	falso caminhão
7	Sun Oct 23 00:46:17 2022,Caminhao Subindo 2	Caminhao_Subindo_0332.png	falso caminhão
8	Sun Oct 23 00:46:45 2022,Caminhao Descendo 3	Caminhao_Descendo_1304.png	verdadeiro caminhão
9	Sun Oct 23 00:47:40 2022,Caminhao Descendo 4	Caminhao_Descendo_3267.png	verdadeiro caminhão
10	Sun Oct 23 00:48:01 2022,Caminhao Subindo 5	Caminhao_Subindo_4012.png	falso caminhão

Fonte: O autor.

Os itens 6,7 e 10 do Quadro 14, apresentaram falsa detecção, sendo considerados como falha do algoritmo.

4.6 Controle de amostragem de precisão do algoritmo LSBP

O controle foi realizado manualmente com base na análise dos frames extraídos do vídeo original, comparando com os dados obtidos pelo programa de testes, conforme descrito no item 4 e os resultados tabulados como apresentado no Quadro 15 e as imagens de registro estão no

Quadro 27.

Quadro 15 - Controle - Teste 01 - LSBP

Item	Tempo,Veiculo	Arquivo	Avaliação
1	Fri Dec 2 10:42:43 2022,Carro Descendo 1	Carro_Descendo_0086.png	Verdadeiro Carro
2	Fri Dec 2 10:42:51 2022,Carro Descendo 2	Carro_Descendo_0132.png	Verdadeiro Carro
3	Fri Dec 2 10:42:57 2022,Carro Descendo 3	Carro_Descendo_0164.png	Verdadeiro Carro
4	Fri Dec 2 10:43:00 2022,Carro Subindo 4	Carro_Subindo_0172.png	Verdadeiro Carro
5	Fri Dec 2 10:43:15 2022,Carro Descendo 5	Carro_Descendo_0250.png	Verdadeiro Carro
6	Fri Dec 2 10:43:21 2022,Caminhao Descendo 1	Caminhao_Descendo_0287.png	Falso Caminhão
7	Fri Dec 2 10:43:28 2022,Caminhao Subindo 2	Caminhao_Subindo_0322.png	Falso Caminhão
8	Fri Dec 2 10:43:38 2022,Caminhao Descendo 3	Caminhao_Descendo_0369.png	Falso Caminhão
9	Fri Dec 2 10:44:00 2022,Caminhao Subindo 4	Caminhao_Subindo_0438.png	Verdadeiro Caminhão
10	Fri Dec 2 10:44:41 2022,Caminhao Subindo 5	Caminhao_Subindo_0555.png	Verdadeiro Caminhão

Fonte: O autor.

Os itens 6,7 e 8 do Quadro 15, apresentaram Oclusão, sendo considerados como falha do algoritmo.

4.7 Controle de amostragem de precisão do algoritmo GSOC

O controle foi realizado manualmente com base na análise dos frames extraídos do vídeo original, comparando com os dados obtidos pelo programa de testes, conforme descrito no item 4 e os resultados tabulados como apresentado no Quadro 16 e as imagens de registro estão no

Quadro 28.

Quadro 16 - Controle - Teste 01 - GSOC

ITEM	Tempo,Veiculo	Arquivo	Avaliação
1	Fri Dec 2 12:52:05 2022,Carro Descendo 1	Carro_Descendo_0087.png	Verdadeiro carro
2	Fri Dec 2 12:52:08 2022,Carro Subindo 2	Carro_Subindo_0118.png	Verdadeiro carro
3	Fri Dec 2 12:52:09 2022,Carro Descendo 3	Carro_Descendo_0132.png	Verdadeiro carro
4	Fri Dec 2 12:52:11 2022,Carro Descendo 4	Carro_Descendo_0165.png	Verdadeiro carro
5	Fri Dec 2 12:52:18 2022,Carro Descendo 5	Carro_Descendo_0250.png	Verdadeiro carro
6	Fri Dec 2 12:52:35 2022,Caminhao Subindo 1	Caminhao_Subindo_0438.png	Verdadeiro caminhão
7	Fri Dec 2 12:52:45 2022,Caminhao Subindo 2	Caminhao_Subindo_0555.png	Verdadeiro caminhão
8	Fri Dec 2 12:53:43 2022,Caminhao Subindo 3	Caminhao_Subindo_1222.png	Verdadeiro caminhão
9	Fri Dec 2 12:53:53 2022,Caminhao Descendo 4	Caminhao_Descendo_1344.png	Verdadeiro caminhão
10	Fri Dec 2 12:54:57 2022,Caminhao Subindo 5	Caminhao_Subindo_2123.png	Verdadeiro caminhão

Fonte: O autor.

O item 8 do Quadro 16, apresentou Oclusão, sendo considerado como uma falha do algoritmo.

5 CONCLUSÕES

Os algoritmos apresentados, mesmo sendo o foco dos testes, o emprego de veículos terrestres, como meio de testes, pode ser empregado para o rastreamento de movimento de pessoas, porém podendo ser adaptados a quaisquer outros objetos.

Os algoritmos kNN, MoG, MoG2, GMG, GSoC e LSBP abordados nesse trabalho são parte da biblioteca OpenCV; que é empregada em muitas soluções comerciais, esta biblioteca se coloca em constante desenvolvimento e agrega valor as soluções que surgem com base em seus algoritmos, como demonstraram os autores referenciados.

A aplicação prática de sistemas de visão para o gerenciamento e monitoramento relacionados ao tráfego e áreas correlatas, assim como apresentado, defendem a utilização conjunta das técnicas de visão computacional, lógica Fuzzy, FES, GA, PSO, ANN entre outras. Algumas destas técnicas foram revisadas e comparado aos sistemas estáticos, os com base em IA forneceram benefícios substanciais para a gestão de tráfego.

Outros trabalhos expuseram os ganhos em diferentes abordagens com o uso de visão computacional em seus trabalhos, contudo assim como (LARA, 2007) afirma, não haver algoritmo que seja robusto o bastante para se adaptar as diversidades das situações ambientais e ambientes não controlados; alertando para o tipo de algoritmo a ser empregado na segmentação; os trabalhos demonstram um grande avanço, assim como nas áreas de conhecimento paralelos à área de visão computacional; no surgimento de hardwares e softwares especializados nas tecnologias utilizadas por estes pesquisadores, como IA, redes neurais, sistema de visão computacional baseadas em CPU, GPU e TPU, além de aplicações em Cloud, quando o processamento Edge não seja necessário ou fundamental para a aplicação requerida. Sistemas diferentes, alguns considerados mais robustos como o *Yolo*, *TensorFlow*, *CUDA*, *Darknet*, *Matlab*, *SimpleCV* e *BoofCV* podem se aproximar da visão biológica ou ganhar em desempenho dos algoritmos tradicionais apresentados nesse trabalho.

Embora não tenham chegado à eficiência desejada, os algoritmos apresentados, demonstram a evolução e o comportamento distintos, embora haja a necessidade de suporte de outras tecnologias para que sejam mais eficientes.

5.1 Respondendo as questões da pesquisa

No contexto de assertividade, os algoritmos como apresentados, não demonstraram acurácia necessária para seu emprego direto em uma solução do tipo apresentada.

Devido ao vídeo utilizado nesse trabalho possuir interferências (vento, iluminação, sombras, outros objetos em movimento e outros), os algoritmos baseados em predição de posição ou estimativa de movimento se adequam melhor por utilizarem ferramentas estatísticas como Filtros de *Kalman* ou ainda Redes *Bayesianas* dinâmicas, para estimar a posição dos objetos no quadro seguinte do vídeo.

Mas isso não foi o suficientemente preciso para executar a contagem de forma assertiva, sendo que nenhum dos algoritmos testados para os grupos CARROS e CAMINHÕES, conseguiram obter resultados satisfatórios (seria esperado algo próximo de 99%).

Contando apenas com o recurso de subtração de *background* e rastreamento, não é possível utilizar estes algoritmos para o gerenciamento do funcionamento de um semáforo como idealizado, pois os resultados demonstram que não há repetibilidade e precisão, o que acarretaria possíveis erros e conseqüentemente acidentes poderiam ocorrer devido ao baixo fator de assertividade.

Por haver oclusão em muitos momentos, fato averiguável através dos frames extraídos, fez com que os algoritmos não contabilizassem de forma assertiva e que para uma aplicação prática e segura, deve ser acrescida de outra tecnologia.

Por esta razão, soluções práticas nesse setor, utiliza-se de redes neurais, aprendizado de máquina e aprendizado profundo, como o *YOLO*, utilizando-se de *DARKNET* com aceleração de hardware com base em GPU como o *CUDA*; obteriam melhores resultados e confiabilidade na gestão semafórica. Mas esse tipo de hardware consumiria mais energia e o custo de aquisição certamente será maior. Toda via, em casos em que não seja necessário uma alta confiabilidade ou até baixa demanda no processamento, os algoritmos apresentados podem se sair muito bem, inclusive se houver processamento posterior, como identificação de placas em sistemas de rodízio municipais de veículos, como o que existe na cidade de São Paulo, por exemplo.

5.2 Considerações finais

Como observado, a utilização apenas de *BackgroundSubtractor* com rastreamento, mas sem o emprego de outra tecnologia, como ML, DL, *CUDA* ou ainda, outra tecnologia baseada em GPU e/ou TPU, não é satisfatória e poderia apresentar desempenho e até confiabilidade o bastante, para ser empregada na gestão semafórica em tempo real.

Porém para outras soluções em que a segurança não seja um fator determinante, ou ainda a precisão na contagem, a solução apresentada pode atender com certa confiabilidade nos dados, desde que não seja em tempo real ou ainda, que se permita determinado erro nos dados, após é claro, uma análise nos valores aplicados nos filtros de forma específica para determinado algoritmo, assim obtendo melhores resultados direcionados ao algoritmo escolhido.

Entende-se que dispositivos de borda, devem ser capazes de processar a quantidade de informações a que se destinam, sem dependerem de servidores, nesse caso, o trabalho focou na análise da precisão e assertividade dos algoritmos, mas há outras necessidades em um dispositivo IOT, como o consumo de energia durante o funcionamento; como é do conhecimento do público, o consumo é proporcional ao uso da CPU e com opções de utilização de GPU, ou ainda de arquiteturas TPU, pode-se ainda ter outra derivação de pesquisa, pois se o algoritmo escolhido demandar mais de uma arquitetura do que de outra, qual seria a melhor escolha se levada em consideração o conceito de sustentabilidade da ODS ou apenas do DOTS?

5.3 Trabalhos futuros

Existem muitos GAP's a serem explorados, muitas opções e variações a serem seguidas, por hora, observa-se:

- Testar e avaliar outros algoritmos em conjunto com os testados, para se obter uma abrangência maior e assim, avaliar e tabular os resultados com mais opções de escolha de algoritmos para IOT ou sistema de CFTV.

- Treinar e obter um conjunto de dados com base nas definições dos veículos conforme o Anexo I da Lei nº 9.503, de 23 de setembro de 1997 do (BRASIL - MINISTÉRIO DA INFRAESTRUTURA, 2021).

- Incluir um levantamento de precisão com base no CDNET, comparando os resultados de identificação dos algoritmos testados em relação a sua precisão de identificação, como por exemplo: falso positivos. Para esse caso, obteve-se 28.405 arquivos de imagens em 56 diretórios diferentes, onde foram obtidos durante os testes comparativos e que podem auxiliar no processo mais aprofundado quanto: falso positivo, falso negativo, verdadeiro positivo e verdadeiro negativo, como sugerido no Quadro 17 por exemplo.

Quadro 17 - Métricas de desempenho sugeridas para avaliação em trabalhos futuros

<p>Taxa de verdadeiro positivo (%) = VP / P</p> <p>Taxa de falso positivo (%) = FP / P</p> <p>Exatidão (%) = VP / P</p> <p>Taxa de falso negativo (%) = FN / P</p> <p>Taxa de verdadeiro negativo (%) = VN / P</p> <p>Precisão (%) = $VP + VN / (VP + FP)$</p> <p>Recall = $VP / (VP + FN)$</p> <p>Medida F1 (FM) = $2 \times ((\text{Precisão} \times \text{Recall}) / (\text{Precisão} + \text{Recall})) = 2VP / (2VP + FP + FN)$</p> <p>VP = Verdadeiro positivo, veículos classificados ou mascarados corretamente</p> <p>FP = Falso Positivo, veículos classificados incorretamente ou mal mascarados</p> <p>P = Número de veículos obtidos na contagem do controle manual</p> <p>VN = Verdadeiro Negativo, ausência de veículos classificados ou não mascarados</p> <p>FN = Falso Negativo, veículos não contabilizados ou não mascarados</p> <p>Recall = Quantidade de verdadeiros positivos encontrados entre toda a verdade fundamental</p>
--

Fonte: O autor

Encontrar e testar dispositivos IoT mais apropriados ao uso em um sistema semafórico com base no requisitos obtidos neste trabalho e na gestão de tráfego urbano de pedestres e veículos; assim, no futuro, construir e testar um sistema capaz de gerenciar o trânsito de veículos e pedestres usando o semáforo controlado por um sistema de visão computacional, inteligência artificial e internet das coisas, tratando-o como um agente autônomo em uma malha de dispositivos que se comunicam em rede, e por fim, realizar a análise dos impactos nos indicadores do tráfego urbano, através de estudo de campo ou similar.

Descobrimos ainda:

Aprendizados e propostas de solução;

- a) Descrever o sistema proposto;
- b) Encontrar as limitações, benefícios e custos;
- c) Realizar os testes em campo e possíveis homologações necessárias para a instalação em um cenário real.
- d) Elencar as tecnologias de IOT disponíveis para uso em visão computacional e IA;
- e) Descrever as barreiras e dificuldades legais e operacionais;
- f) Encontrar a melhor solução em Inteligência artificial no gerenciamento de tráfego urbano de veículos e pedestres;

REFERÊNCIAS

- ALI, Ihtisham *et al.* **Real-Time Vehicle Recognition and Improved Traffic Congestion Resolution**. IEEE 2015 13th International Conference on Frontiers of Information Technology (FIT). Islamabad, Pakistan: [s.n.]. 2015. p. 228-233.
- AN, Sheng-hai; LEE, Byung-Hyug; SHIN, Dong-Ryeol. **A Survey of Intelligent Transportation Systems**. 2011 Third International Conference on Computational Intelligence, Communication Systems and Networks. Bali, Indonesia: IEEE. 2011. p. 332-337. Precisa requisitar acesso ao artigo no ResearchGate.
- ANTONELLO, Ricardo. **Introdução a Visão Computacional com Python e OpenCV**. [S.l.]: [S.n.], 2017. Disponível em: <https://professor.luzerna.ifc.edu.br/ricardo-antonello/wp-content/uploads/sites/8/2017/02/Livro-Introdu%C3%A7%C3%A3o-a-Vis%C3%A3o-Computacional-com-Python-e-OpenCV-3.pdf>. Acesso em: 01 Setembro 2022.
- AUGUSTO, Karen Soares. **Identificação Automática do Grau de Maturação de Pelotas de Minério de Ferro**. PUC-Rio - Pontificia Universidade Católica do Rio de Janeiro. Rio de Janeiro. 2012.
- BERGMARK, Pontus. Intro to Image Processing in OpenCV with Python. **Medium**, 2018. Disponível em: <https://medium.com/@pontus.bergmark/intro-to-image-processing-in-opencv-with-python-1c7f94af18a7>. Acesso em: 20 maio 2022.
- BESL, Paul J.; JAIN, Ramesh C. Three-dimensional object recognition. **ACM Computing Surveys**, Michigan, 17, n. 1, mar. 1985. 75-145. Disponível em: <https://dl.acm.org/doi/10.1145/4078.4081>. Acesso em: 5 abr. 2022.
- BHASKAR, Prem K.; YONG, Suet P. **Image processing based vehicle detection and tracking method**. 2014 International Conference on Computer and Information Sciences (ICCOINS). Kuala Lumpur, Malaysia: [s.n.]. 2014.
- BLACKMORE, W. R. Human software. **Behavior Research Methods & Instrumentation**, 13, n. 4, 1981. 553-570. Disponível em: <http://link.springer.com/10.3758/BF03202066>. Acesso em: 1 abr. 2022.
- BOESCH, Gaudenz. What is OpenCV? The Complete Guide (2023) - viso.ai. **viso.ai**, 2023. Disponível em: <https://viso.ai/computer-vision/opencv/>. Acesso em: 25 Janeiro 2023.
- BRASIL - CONTRAN | CONSELHO NACIONAL DE TRÂNSITO - MINISTÉRIO DAS CIDADES. **Manual Brasileiro de Sinalização de Trânsito - VOLUME V - Sinalização Semafórica**. [S.l.]: [S.n.], 2014. 314 p. ISBN 978-85-7958-078-9. Disponível em: https://www.gov.br/infraestrutura/pt-br/assuntos/transito/arquivos-denatran/educacao/publicacoes/manual_vol_v_-2.pdf. Acesso em: 30 Setembro 2021.
- BRASIL - DATASUS. tabnet.datasus.gov.br. **DATASUS Tecnologia da Informação a Serviço do SUS**, 2022. Disponível em: <http://tabnet.datasus.gov.br/cgi/defthtm.exe?sim/cnv/pext10uf.def>. Acesso em: 5 Outubro 2022.

BRASIL - IBGE. IBGE | Cidades | Brasil | Pesquisa | Frota de veículos | Veículo | 2021. **IBGE**, 2021. Disponível em: <https://cidades.ibge.gov.br/brasil/pesquisa/22/28120?ano=2021>. Acesso em: 05 Março 2022.

BRASIL - IBGE. IBGE | Cidades | Brasil | Pesquisa | Frota de veículos | Veículo | 2006. **IBGE**, 2022. Disponível em: <https://cidades.ibge.gov.br/brasil/pesquisa/22/28120?ano=2006&tipo=grafico>. Acesso em: 18 Janeiro 2023.

BRASIL - MINISTÉRIO DA INFRAESTRUTURA. Frota de Veículos - 2021. **Frota de Veículos - 2021**, 26 Fevereiro 2021. Disponível em: <https://www.gov.br/infraestrutura/pt-br/assuntos/transito/conteudo-denatran/frota-de-veiculos-2021>. Acesso em: 01 Outubro 2021. Publicado em 26/02/2021 16h54 | Atualizado em 28/09/2021 11h10.

C.CHRISTOFHER. Sinopsis Histórica de la IA | Conferencia de Darthmouth 1956. **Sinopsis Histórica de la IA | Conferencia de Darthmouth 1956**, 12 dez. 2021. Disponível em: <https://darthmouthconference.wordpress.com/2014/04/15/sinopsis-historica-de-la-ia/>.

CAVALHEIRO, Eidy Regina Marcílio. **O uso de semáforos inteligentes como instrumento no combate à crise de mobilidade urbana: uma análise da cidade de São Paulo**. Universidade Nove de Julho - UNINOVE. São Paulo, p. 156. 2021.

CHANGEDETECTION.NET. Change Detection. **changedetection.net**, 2015. Disponível em: <http://www.changedetection.net/>. Acesso em: 15 Novembro 2022.

CHAWLA, Nitesh V. *et al.* SMOTE: Synthetic Minority Over-sampling Technique. **Journal of Artificial Intelligence Research**, v. 16, p. 321-357, 01 Junho 2002. Disponível em: <https://www.jair.org/index.php/jair/article/view/10302>. Acesso em: 23 nov. 2022.

CHEUNG, Sen-Ching S.; KAMATH, Chandrika. Robust techniques for background subtraction in urban traffic video. **Proc. SPIE 5308, Visual Communications and Image Processing 2004**, San José, California, United States, 5308, 2004. Disponível em: <https://www.spiedigitallibrary.org/conference-proceedings-of-spie/5308/0000/Robust-techniques-for-background-subtraction-in-urban-traffic-video/10.1117/12.526886.short?SSO=1>. Acesso em: 15 Outubro 2021.

CHILUKURI, Devi M.; YI, Sun; SEONG, Younho. A robust object detection system with occlusion handling for mobile devices. **Computacional Intelligence**, 38, n. 4, 24 Janeiro 2022. 1338-1364. Disponível em: <https://onlinelibrary.wiley.com/doi/full/10.1111/coin.12511>. Acesso em: 25 Novembro 2022.

CHOI, Jaesik. **Realtime On-Road Vehicle Detection with Optical Flows and Haar-Like Feature Detectors**. University of Illinois. Illinois. 2006.

COIFMAN, Benjamin *et al.* A real-time computer vision system for vehicle tracking and traffic surveillance. **Transportation Research Part C: Emerging Technologies**, v. 6, n. 4, p. 271-288, Agosto 1998. Disponível em:

<https://www.sciencedirect.com/science/article/pii/S0968090X98000199>. Acesso em: 05 Abril 2021.

DEDEOGLU, Yigithan. **Moving object detection, tracking and classification for smart video surveillance**. [S.l.]. 2004.

ELGAMMAL, Ahmed; HARWOOD, David; DAVIS, Larry. **Non-parametric Model for Background Subtraction**. European Conference on Computer Vision - ECCV 2000: Computer Vision ECCV 2000. [S.l.]: [s.n.]. 2000. p. 751-767.

EMBRATEL. As quatro eras do reconhecimento facial e seu efeito na privacidade. **Próximo Nível**, 13 dez. 2021. Disponível em: <https://proximonivel.embratel.com.br/as-quatro-eras-do-reconhecimento-facial-e-seu-efeito-na-privacidade/>.

ENCYCLOPEDIA BRITANNICA ONLINE. Wikipedia. **Wikipedia.org**. Disponível em: <https://pt.wikipedia.org/wiki/BLOB>. Acesso em: 28 Setembro 2021.

FATHY, Mahmood; SIYAL, Mohammed Y. An image detection technique based on morphological edge detection and background differencing for real-time traffic analysis. **Pattern Recognition Letters**, v. 16, n. 12, p. 1321-1330, Dezembro 1995.

FILHO, Ogê M.; NETO, Hugo V. **Processamento Digital de Imagens**. Rio de Janeiro: Brasport, 1999. ISBN ISBN 8574520098.

FORBES. Relembre os 10 piores congestionamentos - Forbes. **Forbes Brasil**, 15 Março 2017. Disponível em: <https://forbes.com.br/listas/2017/03/10-piores-congestionamentos-da-historia/>. Acesso em: 01 Setembro 2022.

FREUND, Yoav; SCHAPIRE, Robert E. A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting. **Pennsylvania State University**, 20 setembro 1985. 34. Disponível em: <https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.56.9855>.

FU, K. S.; GONZALEZ, R. C.; LEE, C.S.G. Sensing. In: _____ **Robotics: Control, Sensing, Vision and Intelligence**. [S.l.]: McGraw-Hill Book Company, 1987. Cap. 6, p. 266-295. Disponível em: http://enr.case.edu/merat_francis/eecs490f06/References/Fu/Fu_Ch6.pdf. Acesso em: 3 Maio 2022.

GEEKFORGEEKS. OpenCV – Overview | GeekforGeeks. **GeekforGeeks**, 2022. Disponível em: <https://www.geeksforgeeks.org/opencv-overview/>. Acesso em: 20 Dezembro 2022.

GEEKSFORGEEKS. Background subtraction – OpenCV - GeeksforGeeks. **GeeksforGeeks**, 2020. Disponível em: <https://www.geeksforgeeks.org/background-subtraction-opencv/>. Acesso em: 15 Abril 2022.

GEEKSFORGEEKS.ORG. Face detection using Cascade Classifier using OpenCV-Python. **geeksforgeeks.org**, 2023. Disponível em: <https://www.geeksforgeeks.org/face-detection-using-cascade-classifier-using-opencv-python/>. Acesso em: 10 Janeiro 2023.

GIL, Antonio C. **Métodos e Técnicas de Pesquisa Social**. 6ª. ed. São Paulo: Atlas, 2008. 49 p. ISBN ISBN 978-85-224-5142-5.

GIOVANINI, Adenilson. Processamento de Imagem: As 3 Etapas? **Professor Adenilson Giovanini**, 2022. Disponível em: <https://adenilsongiovanini.com.br/blog/processamento-de-imagem-as-3-etapas/>. Acesso em: 15 Dezembro 2022.

GODBEHERE, Andrew B.; MATSUKAWA, Akihiro; GOLDBERG, Ken. **Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation**. 2012 American Control Conference (ACC). [S.l.]: [s.n.]. 2012.

GONZALEZ, Rafael C.; WOODS, Richard E. **Processamento de imagens digitais**. 3. ed. São Paulo: Editora Blucher, 2007. 528 p. ISBN 8521217269, 9788521217268. Livro impresso físico doado pelo amigo Willians Pereira Dutra.

GRANATYR, Jones; VARGAS, Dalton Luiz; , IA Expert Academy. Udemy. **Udemy**, 06 Dezembro 2022. Disponível em: <https://www.udemy.com/course/rastreamento-objetos-python-opencv/>.

GREESHMA , C. A. *et al.* Traffic control using computer vision. **IJARIT - International Journal of Advance Research, Ideas And Innovations In Technology**, 5, n. 1, 2019. 517-520. Disponível em: <https://www.ijarit.com/manuscripts/v5i1/V5I1-1338.pdf>. Acesso em: 05 Fevereiro 2022.

GUO, Lili; XU, Dan ; QIANG, Zhenping. **Background Subtraction Using Local SVD Binary Pattern**. 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). Las Vegas, NV, USA: IEEE. 01 Junho 2016. p. 86-94.

HACK THE DEVELOPER. Background-subtraction OpenCV Python. **hackthedeveloper.com**, 2022. Disponível em: <https://hackthedeveloper.com/background-subtraction-opencv-python/>. Acesso em: 15 Março 2022.

HAMILTON, Dale *et al.* **kNN vs. SVM: A Comparison of Algorithms**. Proceedings of the Fire Continuum Conference. Missoula: [s.n.]. 2018. p. 95-109. U.S. Forest Service RMRS P-78. 2020.

HAN, Sungji; HAN, Youngjoon; HAHN, Hernsoo. Vehicle detection method using Haar-like feature on real time system. **World Academy of Science, Engineering and Technology**, 59, 21 Novembro 2009. Disponível em: https://www.researchgate.net/publication/255633155_Vehicle_Detection_Method_using_Haar-like_Feature_on_Real_Time_System. Acesso em: 23 nov. 2022. Precisa solicitar permissão de acesso ao documento em ResearchGate.

HAWI, Roxanne ; OKEYO, George ; KIMWELE, Michael. Techniques for Smart Traffic Control: An In-depth Review. **International Journal of Computer Applications Technology and Research**, v. 4, n. 7, p. 566 - 573, 2015. ISSN ISSN: 2319-8656. Disponível em: <https://ijcat.com/archives/volume4/issue7/ijcatr04071014.pdf>. Acesso em: 23 nov. 2022.

HERNÁNDEZ-HERNÁNDEZ, José Luis *et al.* **Search for Optimum Color Space for the Recognition of Oranges in Agricultural Fields**. CITI 2017, CCIS 749. [S.l.]: Springer International Publishing. 2017. p. 296-307.

HERRERO, Sonsoles; BESCÓS, Jesús. **Background Subtraction Techniques: Systematic Evaluation and Comparative Analysis**. International Conference on Advanced Concepts for Intelligent Vision Systems. [S.l.]: Springer. 2009. p. 33-42.

IFTIKHAR, Zubair; DISSANAYAKE, Prashan; VIAL, Peter. **Computer Vision Based Traffic Monitoring System for Multi-track Freeways**. International Conference on Intelligent Computing. [S.l.]: Springer International Publishing Switzerland. 2014. p. 339-349.

INTERNATIONAL TRANSPORT FORUM. ITF. **International Transport Forum, 2022**. Disponível em: <https://www.itf-oecd.org/sites/default/files/docs/irtad-road-safety-annual-report-2021.pdf>. Acesso em: 5 Outubro 2022.

JÄHNE, Bernd *et al.* **Computer Vision and Applications: A Guide for Students and Practitioners**. Orlando: Academic Press, 2000. Disponível em: <https://library.uoh.edu.iq/admin/ebooks/26305-bernd-jahne---computer-vision-and-applications.pdf>. Acesso em: 12 Dezembro 2021.

JAHNE, Bernd; HAUBECKER, Horst. **Computer Vision and Applications: A Guide for Students and Practitioners**. [S.l.]: Academic Press, 2000. 702 p. ISBN 0-12-379777-2.

JAISWAL, Abhishek. Object Detection Using Haar Cascade: OpenCV. **Analytics Vidhya**, 2022. Disponível em: <https://www.analyticsvidhya.com/blog/2022/04/object-detection-using-haar-cascade-opencv/>. Acesso em: 5 Novembro 2022.

JIMÉNEZ-BRAVO, Diego M. *et al.* Multi-object tracking in traffic environments: A systematic literature review. **Neurocomputing**, n. 494, 20 Abril 2022. 43-55. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0925231222004672?via%3Dihub>. Acesso em: 21 Agosto 2022.

KAEWTRAKULPONG, P.; BOWDEN, R. An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection. In: REMAGNINO, Paolo, *et al.* **Video-Based Surveillance Systems: Computer Vision and Distributed Processing**. 1. ed. New York: Springer Science+Business Media, v. 1, 2002. Cap. 11, p. 135-144. ISBN ISBN: 978-1-4615-0913-4. Disponível em: https://link.springer.com/chapter/10.1007/978-1-4615-0913-4_11. Acesso em: 12 ago. 2022.

KANUNGO, Anurag ; SHARMA, Ayush ; SINGLA, Chetan. **Smart traffic lights switching and traffic density calculation using video processing**. 2014 Recent Advances in Engineering and Computational Sciences (RAECS). [S.l.]: [s.n.]. 2014.

KHADER, Salima M. Fuzzy Mathematics (Fuzzy Sets). **Scientific Journal of Faculty of Education, Misurata University-Libya**, Líbia, v. 1, n. 17, Março 2021. ISSN 2710- 4141. Disponível em: <https://www.misuratau.edu.ly/journal/edu/upload/file/E4.pdf>. Acesso em: 21 nov. 2022.

KOLLER, Daphne; DANILIDIS, Kostas; NAGEL, Hans H. Model-based object tracking in monocular image sequences of road traffic scenes. **INTERNATIONAL JOURNAL OF COMPUTER VISION** 10:3, p. 257-281, 1993. Disponível em: <https://link.springer.com/article/10.1007/BF01539538>. Acesso em: 03 jan. 2022.

KRISHNANI, Pramod; SUL, Dongun; ANYANWU, Chima. **Automatic Traffic Light Control System**. Sacramento: [S.n.], 2008. 12 p. Disponível em: <https://citeseerx.ist.psu.edu/pdf/674dc11aa6faa892df8b805e520e4d9a4d21c8b3>. Acesso em: 28 Setembro 2021.

LARA, Arnaldo Câmara. **Segmentação de movimento usando morfologia matemática**. Proceedings of Simpósio Brasileiro de Computação Gráfica (SIBGRAPI). Belo Horizonte, Minas Gerais: [s.n.]. Novembro 2007.

LEARNOPENCV.COM. Contour Detection using OpenCV (Python/C++). **LearnOpenCV.com**, 2022. Disponível em: <https://learnopencv.com/contour-detection-using-opencv-python-c/>. Acesso em: 18 Março 2022.

LIU, Wei *et al.* SSD: Single Shot MultiBox Detector. **European Conference on Computer Vision 2016: Computer Vision – ECCV**, n. 1, p. 21-37, 17 Setembro 2016. ISSN 978-3-319-46448-0. Disponível em: https://link.springer.com/chapter/10.1007/978-3-319-46448-0_2. Acesso em: 28 Setembro 2021.

MARCOMINI, Leandro Arab; CUNHA, André Luiz. A Comparison between Background Modelling Methods for Vehicle Segmentation in Highway Traffic Videos. **ARXIV - Cornell University**, 5 Outubro 2018. 12. Disponível em: <https://arxiv.org/ftp/arxiv/papers/1810/1810.02835.pdf>. Acesso em: 10 Junho 2022.

MARCONI, Marina de A.; LAKATOS, Eva M. **Fundamentos de metodologia científica**. 5ª ed. São Paulo: Atlas, 2003. ISBN ISBN 85-224-3397-6.

MARR, David. **From computational theory to psychology and neurophysiology - a case study from vision**. Massachusetts Institute of Technology. [S.l.], p. 29. 1976.

MCIVOR, Alan M. **Background Subtraction Techniques**. Proc. of Image and Vision Computing. Auckland: [s.n.]. 2000.

MITTAL, Aditya. Haar Cascades, Explained. **Medium.com**, 2020. Disponível em: <https://medium.com/analytics-vidhya/haar-cascades-explained-38210e57970d>. Acesso em: 5 Outubro 2022.

MITTAL, Pardeep ; SINGH, Yashpal; SHARMA, Yogesh. Analysis of Dynamic Road Traffic Congestion Control (DRTCC) Techniques. **American Journal of Engineering Research (AJER)**, v. 4, n. 10, p. 40-47, Outubro 2015. ISSN e-ISSN: 2320-0847 p-ISSN : 2320-0936. Disponível em: [http://www.ajer.org/papers/v4\(10\)/G041040047.pdf](http://www.ajer.org/papers/v4(10)/G041040047.pdf). Acesso em: 23 nov. 2022.

MOESLUND, Thomas B.; HILTON, Adrian; KRUGER, Volker. A survey of advances in vision-based human motion capture and analysis. **Computer Vision and Image Understanding**, 2006. 90-126.

MURZOVA, Anastasia. Background Subtraction with OpenCV and BGS Libraries. **learnopencv.com**, 2021. Disponível em: <https://learnopencv.com/background-subtraction-with-opencv-and-bgs-libraries/>. Acesso em: 20 Abril 2022.

NEVES, Samuel Clayton Maciel ; PELAES, Evaldo Gonçalves. Estudo e implementação de técnicas de segmentação de imagens. **Revista Virtual de Iniciação Acadêmica da UFPA**, 1, n. 2, Julho 2001. 1-11. Disponível em:

https://www.academia.edu/2125627/Estudo_e_implementa%C3%A7%C3%A3o_de_t%C3%A9cnicas_de_segmenta%C3%A7%C3%A3o_de_imagens. Acesso em: 01 fev. 2022.

NVIDIA. Nvidia. **nvidia.com**, 15 nov. 2022. Disponível em: <https://www.nvidia.com/pt-br/glossary/data-science/computer-vision/>.

OPENCADD. OpenCADD. **OpenCADD**, 12 dez. 2021. Disponível em:

https://www.opencadd.com.br/wp-content/uploads/2019/01/dartmouth_summer_project.pdf.

OPENCV.ORG (10). OpenCV: cv:bgsegm:BackgroundSubtractorLSBP Class Reference.

OpenCV.org, 02 mar. 2022. Disponível em:

https://docs.opencv.org/4.5.5/de/d4c/classcv_1_1bgsegm_1_1BackgroundSubtractorLSBP.html.

OPENCV.ORG (11). OpenCV: Motion Analysis. **OpenCV.org**, 2022. Disponível em:

https://docs.opencv.org/4.x/de/de1/group__video__motion.html#ga2beb2dee7a073809ccec60f145b6b29c. Acesso em: 08 Maio 2022.

OPENCV.ORG (2). OpenCV: Improved Background-Foreground Segmentation Methods.

OpenCV.org, 2022. Disponível em:

https://docs.opencv.org/4.x/d2/d55/group__bgsegm.html#ga17d9525d2ad71f74d8d29c2c5e11903d. Acesso em: 08 Maio 2022.

OPENCV.ORG (4). OpenCV: Understanding k-Nearest Neighbour | OpenCV.org.

OpenCV.org, 2022. Disponível em:

https://docs.opencv.org/3.4/d5/d26/tutorial_py_knn_understanding.html. Acesso em: 11 Março 2022.

OPENCV.ORG (5). OpenCV: cv:BackgroundSubtractorKNN Class Reference. **OpenCV.org**, 25 mar. 2022. Disponível em:

https://docs.opencv.org/4.5.5/db/d88/classcv_1_1BackgroundSubtractorKNN.html.

OPENCV.ORG (5). OpenCV: cv:bgsegm:BackgroundSubtractorMOG Class Reference.

OpenCV.org, 15 mar. 2022. Disponível em:

https://docs.opencv.org/4.5.5/d6/da7/classcv_1_1bgsegm_1_1BackgroundSubtractorMOG.html.

OPENCV.ORG (6). OpenCV: cv:BackgroundSubtractorMOG2 Class Reference.

OpenCV.org, 25 mar. 2022. Disponível em:

https://docs.opencv.org/4.5.5/d7/d7b/classcv_1_1BackgroundSubtractorMOG2.html.

OPENCV.ORG (7). OpenCV: cv:bgsegm:BackgroundSubtractorGMG Class Reference.

OpenCV.org, 30 mar. 2022. Disponível em:

https://docs.opencv.org/4.5.5/d1/d5c/classcv_1_1bgsegm_1_1BackgroundSubtractorGMG.html.

OPENCV.ORG (8). OpenCV: cv:bgsegm:BackgroundSubtractorCNT Class Reference. **OpenCV.org**, 26 mar. 2022. Disponível em: https://docs.opencv.org/4.5.5/de/dca/classcv_1_1bgsegm_1_1BackgroundSubtractorCNT.html.

OPENCV.ORG (9). OpenCV: cv:bgsegm:BackgroundSubtractorGSOC Class Reference. **OpenCV.org**, 25 mar. 2022. Disponível em: https://docs.opencv.org/4.5.5/d4/dd5/classcv_1_1bgsegm_1_1BackgroundSubtractorGSOC.html#details.

OPENCV.ORG (A). About OpenCV. **OpenCV.org**, 02 out. 2022. Disponível em: <https://opencv.org/about/>.

OPENCV.ORG (B). Home - OpenCV. **OpenCV.org**, 26 nov. 2022. Disponível em: <https://opencv.org/about/>. Acesso em: 10 Janeiro 2023.

OPENCV.ORG (D). OpenCV: Contour Features. **OpenCV**, 2021. Disponível em: https://docs.opencv.org/3.4/dd/d49/tutorial_py_contour_features.html. Acesso em: 2 Julho 2022.

OPENCV.ORG (XX). OPENCV: Cascade Classifier. **opencv.org**, 2022. Disponível em: https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html. Acesso em: 21 Março 2022.

OPENCV.ORG (XXX). OpenCV: cv:BackgroundSubtractor Class Reference. **OpenCV.org**, 25 mar. 2022. Disponível em: https://docs.opencv.org/4.5.5/d7/df6/classcv_1_1BackgroundSubtractor.html.

OUSSAMA, Maamri. Change Detection 2014 - A video database for testing change detection algorithms. **Kaggle**, 2019. Disponível em: <https://www.kaggle.com/datasets/maamri95/cdnet2014>. Acesso em: 12 Outubro 2022.

PACKT. What can you do with OpenCV? **packtpub.com**, 2022. Disponível em: <https://subscription.packtpub.com/book/data/9781785280948/1/ch011v11sec10/what-can-you-do-with-opencv>. Acesso em: 10 Dezembro 2022.

PATEL, N.; RANGANATHAN, N. IDUTC: an intelligent decision-making system for urban traffic-control applications. **IEEE Transactions on Vehicular Technology**, New York, NY, 50, n. 3, Maio 2001. 816-829. Disponível em: <https://ieeexplore.ieee.org/document/933315>. Acesso em: 28 Julho 2021.

PEÑA-GONZÁLEZ, Raúl H.; NUÑO-MAGANDA, Marco A. **Computer vision based real-time vehicle tracking and classification system**. 2014 IEEE 57th International Midwest Symposium on Circuits and Systems (MWSCAS). College Station, TX, USA: [s.n.]. 2014. p. 679-682.

PEREZ, Yuri. **Simulação de interrupções de semáforos em sistemas de transporte com o uso da teoria de redes complexas**. Universidade Nove de Julho - UNINOVE. São Paulo, p. 78. 2021.

PICCARDI, Massimo. **Background subtraction techniques: a review**. 2004 IEEE International Conference on Systems, Man and Cybernetics. The Hague, Netherlands: IEEE. 2004.

PINHO, Raquel R.; TAVARES, João Manuel R. S.; CORREIA, Miguel V. **SEGUIMENTO DE OBJECTOS EM VISÃO COMPUTACIONAL USANDO MÉTODOS ESTOCÁSTICOS**. Congreso de Métodos Numéricos en Ingeniería 2005. Granada, 4 a 7 de Julio, 2005: [s.n.]. 2005. p. 16.

PÚBLICO. Inteligência artificial faz 50 anos | Ciências. **PÚBLICO**, 15 dez. 2021. Disponível em: <https://www.publico.pt/2006/07/09/jornal/inteligencia-artificial-faz-50-anos-87953>.

PYIMAGE SEARCH. Start Here with Computer Vision, Deep Learning, and OpenCV. **pyimagesearch.com**, 2022. Disponível em: <https://pyimagesearch.com/start-here/>. Acesso em: 23 Dezembro 2022.

QURESHI, Kashif N.; ABDULLAH, Abdul H. A Survey on Intelligent Transportation Systems. **Middle-East Journal of Scientific Research**, 15, 01 jan. 2013. 629-642. Disponível em: https://www.researchgate.net/publication/257367335_A_Survey_on_Intelligent_Transportation_Systems. Acesso em: 23 nov. 22.

REDE NOSSA SÃO PAULO. Congestionamento de São Paulo, SP - [redesocialdecidades.org.br](https://www.redesocialdecidades.org.br). **Rede Social Brasileira por Cidades Justas e Sustentáveis**, 2012. Disponível em: <https://www.redesocialdecidades.org.br/br/SP/sao-paulo/congestionamento>. Acesso em: 04 Outubro 2022.

REDE SOCIAL BRASILEIRA POR CIDADES JUSTAS E SUSTENTÁVEIS (A). Acidente com automóvel de São Paulo, SP. **redesocialdecidades.org.br**, 2022. Disponível em: <https://www.redesocialdecidades.org.br/br/SP/sao-paulo/acidente-com-automovel>. Acesso em: 20 Dezembro 2022.

REDE SOCIAL BRASILEIRA POR CIDADES JUSTAS E SUSTENTÁVEIS (B). Acidente com bicicleta de São Paulo, SP. **redesocialdecidades.org.br**, 2022. Disponível em: <https://www.redesocialdecidades.org.br/br/SP/sao-paulo/acidente-com-bicicleta>. Acesso em: 20 Dezembro 2022.

REDE SOCIAL BRASILEIRA POR CIDADES JUSTAS E SUSTENTÁVEIS (C). Acidente com motocicleta de São Paulo, SP. **redesocialdecidades.org.br**, 2022. Disponível em: <https://www.redesocialdecidades.org.br/br/SP/sao-paulo/acidente-com-motocicleta>. Acesso em: 20 Dezembro 2022.

REDE SOCIAL BRASILEIRA POR CIDADES JUSTAS E SUSTENTÁVEIS (D). Mortes por atropelamento - São Paulo. **redesocialdecidades.org.br**, 2022. Disponível em: <https://www.redesocialdecidades.org.br/br/SP/sao-paulo/mortes-por-atropelamento>. Acesso em: 20 Dezembro 2022.

REDE SOCIAL BRASILEIRA POR CIDADES JUSTAS E SUSTENTÁVEIS (E). São Paulo, SP. **redesocialdecidades.org.br**, 2022. Disponível em: <https://www.redesocialdecidades.org.br/br/SP/sao-paulo>. Acesso em: 10 Dezembro 2022.

- REDE SOCIAL BRASILEIRA POR CIDADES JUSTAS E SUSTENTÁVEIS (F). Atropelamentos - São Paulo. **redesocialdecidades.org.br**, 2022. Disponível em: <https://www.redesocialdecidades.org.br/br/SP/sao-paulo/atropelamentos>. Acesso em: 20 Dezembro 2022.
- REDE SOCIAL BRASILEIRA POR CIDADES JUSTAS E SUSTENTÁVEIS. Dados abertos. **Rede Social Brasileira por Cidades Justas e Sustentáveis**, 2023. Disponível em: <https://www.redesocialdecidades.org.br/dados-abertos>. Acesso em: 27 Janeiro 2023.
- REDESOCIALDECIDADES.ORG.BR (D). Acidentes de trânsito - São Paulo. **redesocialdecidades.org.br**, 2022. Disponível em: <https://www.redesocialdecidades.org.br/br/SP/sao-paulo/acidentes-de-transito>. Acesso em: 20 Dezembro 2022.
- REDMON, J. *et al.* You only look once: Unified, real-time object detection. in **Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.**, Junho 2016. 779–788. Disponível em: <https://sci-hub.se/10.1109/CVPR.2016.91>. Acesso em: 23 nov. 2022.
- REDMON, Joseph; FARHADI, Ali. **Yolo9000**: Better, Faster, Stronger. IEEE International Conference Computacional Vison Pattern Recognit., Jul. 2017. [S.l.]: [s.n.]. Julho 2017. p. 7263-7271.
- ROSEBROCK, Adrian. Local Binary Patterns with Python & OpenCV. **Pyimagesearch**, 2015. Disponível em: <https://pyimagesearch.com/2015/12/07/local-binary-patterns-with-python-opencv/>. Acesso em: 02 Setembro 2022.
- SAMSONOV, Vladislav. `opencv_contrib/modules/bgsegm/src/bgfg_gsoc.cpp`. **github.com**, 2020. Disponível em: https://github.com/opencv/opencv_contrib/blob/master/modules/bgsegm/src/bgfg_gsoc.cpp. Acesso em: 14 Março 2022.
- SEEG. SEEG Brasil. **SEEG - Sistema de Estimativa de Emissões e Remoções de Gases de Efeito Estufa, Observatório do Clima**, 2020. Disponível em: <http://seeg.eco.br>. Acesso em: 4 Outubro 2022.
- SEENOUVONG, Nilakorn *et al.* A computer vision based vehicle detection and counting system. **2016 - 8th International Conference on Knowledge and Smart Technology (KST), IEEE**, Fevereiro 2016. 224-227. Disponível em: https://www.researchgate.net/publication/301709907_A_computer_vision_based_vehicle_detection_and_counting_system. Acesso em: 23 nov. 2022. Necessita solicitar aos autores o acesso ao arquivo completo.
- SIEBEL, Nils T.; MAYBANK, Stephen J. The ADVISOR Visual Surveillance System, 2004.
- SILVA, Fausto Carlos Filipe da ; SISWANTO, Diky ; ISTIADI, I.. **Design Prototype of Traffic Light Control System using HSV Colour Segmentation**. Evolution of Information, Communication and Computing Systems (EICCS). [S.l.]: [s.n.]. 2022. p. 80-91.

SINGH, Y. P.; MITTAL, Pradeep. Analysis and Designing of Proposed Intelligent Road Traffic Congestion Control System with Image Mosaicking Technique. **International Journal of IT, Engineering and Applied Sciences Research (IJIEASR)**, v. 2, n. 4, p. 27-31, Abril 2013. ISSN 2319-4413. Disponível em:

<http://www.irjournals.org/ijieasr/Apr2013/7.pdf>. Acesso em: 14 mar. 2021.

SOLAWETZ, Jacob. Occlusion Techniques in Computer Vision. **roboflow**, 2020. Disponível em: <https://blog.roboflow.com/occlusion-computer-vision/>. Acesso em: 15 Dezembro 2022.

STACKOVERFLOW. Image Processing: What are occlusions? - StackOverflow.

StackOverflow, 2018. Disponível em: <https://stackoverflow.com/questions/2764238/image-processing-what-are-occlusions>. Acesso em: 15 Dezembro 2022.

STRATMANN, Lukas. color.lukas-stratmann. **color.lukas-stratmann**, 07 Março 2022.

Disponível em: <http://color.lukas-stratmann.com/color-systems/rgb.html#:~:text=The%20RGB%20color%20model%20can%20be%20understood%20as,b%20are%20the%20color%27s%20intensities%20for%20each%20channel>. Acesso em: 10 Outubro 2022.

TAN, T. N. ; SULLIVAN, G. D.; BAKER, K. D. **Fast Vehicle Localisation and Recognition Without Line Extraction and Matching**. Proceedings of the British Machine Vision Conference. [S.l.]: BMVA Press. 1994. p. 85-94.

TECMUNDO. A história da Internet: a década de 1990 [infográfico] - TecMundo.

TecMundo, 12 dez. 2021. Disponível em: <https://www.tecmundo.com.br/infografico/10054-a-historia-da-internet-a-decada-de-1990-infografico-.htm>.

THAKOR, Aniruddh; ASKHEDKAR, Anjali. Real time object tracking to remove occlusion using OpenCV. **International Journal of Scientific & Engineering Research**, 5, n. 5, Maio 2014. 1420-1424. Disponível em: <https://www.ijser.org/researchpaper/Real-time-object-tracking-to-remove-occlusion-using-OpenCV.pdf>. Acesso em: 08 Dezembro 2022.

TSUCHIYA, Haruki *et al.* **A Method of Data Augmentation for Classifying Road Damage Considering Influence on Classification Accuracy**. 23rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems. [S.l.]: Elsevier B.V. 2019. p. 1449-1458.

VIAS SEGURAS. Estatísticas nacionais de acidentes de trânsito | Vias seguras. **Vias Seguras**, 28 nov. 2022. Disponível em: http://www.vias-seguras.com/os_acidentes/estatisticas/estatisticas_nacionais.

VIOLA, Paul; JONES, Michael. Rapid Object Detection using a Boosted Cascade of Simple Features. **Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001**, Kauai, HI, USA, p. I-511~I-518, 08-14 Dezembro 2001. ISSN ISBN:0-7695-1272-0 | ISSN: 1063-6919. Disponível em: <https://cs.colby.edu/maxwell/courses/cs397-vision/F07/papers/viola-Faces-cvpr01.pdf>. Acesso em: 16 Março 2022.

VIOLA, Paul; JONES, Michael J.; SNOW, Daniel. Detecting Pedestrians Using Patterns of Motion and Appearance. **Computer Science - International Journal of Computer Vision**,

13 Outubro 2003. ISSN 0-7695-1950-4. Disponível em:
<https://ieeexplore.ieee.org/document/1238422>. Acesso em: 10 nov. 2021.

WANG, Ce ; BRANDSTEIN, Michael S.. A hybrid real-time face tracking system. **Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98**, Seattle, 1998.

WANG, Kelvin C. P.; HOU, Zhiqiong; GONG, Weiguo. Automated Road Sign Inventory System Based on Stereo Vision and Tracking. **2010 Computer-Aided Civil and Infrastructure Engineering**, 25, Agosto 2010. 468-477. Disponível em:
https://www.researchgate.net/publication/220373574_Automated_Road_Sign_Inventory_System_Based_on_Stereo_Vision_and_Tracking. Acesso em: 23 nov. 2022. Precisa solicitar acesso ao documento no ResearchGate.

WANG, Yi *et al.* CDnet 2014: An Expanded Change Detection Benchmark Dataset. **IEEE CVPR Change Detection workshop**, 4 Junho 2014. 8. Disponível em:
https://www.researchgate.net/publication/278806972_CDnet_2014_An_Expanded_Change_Detection_Benchmark_Dataset?enrichId=rgreq-a1260de3f78fb8ca6edc7db204d4b217-XXX&enrichSource=Y292ZXJQYWdlOzI3ODgwNjk3MjtBUzoyODk5NjUxOTI4MjY4ODBAMTQ0NjE0NDQ5NzUzMA%3D%3D&el=1. Acesso em: 01 Outubro 2021.

WANG, Yi *et al.* **CDnet 2014: An Expanded Change Detection Benchmark Dataset**. 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops. Columbus, OH, USA: IEEE. 2014.

WANGENHEIM, Aldo von. **Visão Computacional**. São Carlos: [s.n.]. 2005.

WEI, Peter *et al.* **City-Scale Vehicle Tracking and Traffic Flow Estimation using Low Frame-Rate Traffic Cameras**. 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and the 2019 International Symposium on Wearable Computers. [S.l.]: ACM. 2019. p. 9.

WORLD HEALTH ORGANIZATION. **Global Status Report On Road Safety 2018**. WHO - World Health Organization. [S.l.], p. 424. 2018. (978-92-4-156568-4).

WORLD HEALTH ORGANIZATION. WHO | Death on the roads. **extranet.who.int**, 2018. Disponível em: <https://extranet.who.int/roadsafety/death-on-the-roads/#ticker/pedestrians>. Acesso em: 5 Outubro 2022.

ZEEVI, Sagi. GitHub: sagi-z/BackgroundSubtractorCNT. **GitHub**, 24 nov. 2022. Disponível em: <https://github.com/sagi-z/BackgroundSubtractorCNT>.

ZINCHENKO, Vladyslav *et al.* Computer Vision in Control and Optimization of Road Traffic. **2020 IEEE Third International Conference on Data Stream Mining & Processing**, Lviv, Ukraine, 21-25 Agosto 2020. 249-254. Disponível em:
<https://ieeexplore.ieee.org/document/9204329>. Acesso em: 10 Setembro 2022.

ZIVKOVIC, Zoran. **Improved Adaptive Gaussian Mixture Model for Background Subtraction**. Conference: Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International. [S.l.]: [s.n.]. 2004. p. 28-31.

ZIVKOVIC, Zoran; HEIJDEN, Ferdinand van der. Efficient adaptive density estimation per image pixel for the task of background subtraction. **Pattern recognition letters**, v. 27, n. 7, p. 773-780, Maio 2006. Disponível em:

<https://www.sciencedirect.com/science/article/pii/S0167865505003521>. Acesso em: 25 mar. 2022.

GLOSSÁRIO

BLOB	<p>BLOB (Binary Large Object, basic large object, BLOB ou BLOB, que significa objeto grande binário ou objeto grande básico), é uma coleção de dados binários armazenados como uma única entidade em um sistema de gerenciamento de banco de dados. São geralmente objetos de imagem, áudio ou outros objetos multimídia, apesar de algumas vezes o código binário executável ser armazenado como um BLOB. (ENCYCLOPEDIA BRITANNICA ONLINE)</p>
BoofCV	<p>BoofCV é uma biblioteca de código aberto escrita especificamente para visão computacional em tempo real. Foi lançado sob uma licença Apache 2.0 para uso acadêmico e comercial. Estas são opções para vários ramos de CV no BoofCV, incluindo processamento de imagem de baixo nível, detecção e rastreamento de recursos, calibração de câmera etc. Alguns dos pacotes no BoofCV incluem funções de processamento de imagem com funções de processamento de imagem que operam em pixels, Visão geométrica para extração recursos de imagem usando geometria 2D e 3D, calibração que tem funções para determinar os parâmetros intrínsecos e extrínsecos da câmera, reconhecimento para reconhecer objetos visuais complicados etc.</p> <p>Link: http://boofcv.org/</p>
CUDA	<p>CUDA ou <i>Compute Unified Device Architecture</i> é uma plataforma de computação paralela criada pela Nvidia e lançada em 2007. É usada por engenheiros de software para processamento de propósito geral usando a unidade de processamento gráfico habilitada para CUDA ou GPU; também possui a biblioteca <i>Nvidia Performance Primitives</i> que contém várias funções para processamento de imagem, sinal e vídeo. Algumas outras bibliotecas e coleções incluem: GPU4Vision, OpenVIDIA para algoritmos de visão computacional populares em CUDA, MinGPU que é uma biblioteca GPU mínima para Visão computacional etc. As linguagens suportadas são: C, C++, Fortran, MATLAB, Python etc. enquanto usando CUDA.</p> <p>Link: https://developer.nvidia.com/cuda-zone</p>
DOTS	Desenvolvimento Orientado ao Transporte Sustentável
IOT	Internet of Things (Internet das coisas)

IOV Internet of Vehicles (Internet dos veículos)

LAN Local Area network

ODS Objetivos de Desenvolvimento Sustentável

OPENCV OpenCV (Open Source Computer Vision Library) é uma biblioteca de visão computacional de código aberto que contém muitas funções diferentes para visão computacional e aprendizado de máquina. Foi criado pela Intel e originalmente lançado em 2000 e tem muitos algoritmos diferentes relacionados à visão computacional que podem realizar uma variedade de tarefas, incluindo detecção e reconhecimento facial, identificação de objetos, monitoramento de objetos em movimento, rastreamento de movimentos de câmera, rastreamento de movimentos oculares, extração de 3D modelos de objetos, criando uma sobreposição de realidade aumentada com um cenário, reconhecendo imagens semelhantes em um banco de dados de imagens, etc. Possui interfaces para C++, Java, Python, MATLAB, entre outras e suporta sistemas operacionais como: Windows, Android, Mac OS, Linux etc.

Link: <https://opencv.org/>

SRAM Static Random-Access Memory

SimpleCV é uma estrutura de visão computacional de código aberto que pode ser usada para construir vários aplicativos de visão computacional. Pode-se usar várias bibliotecas avançadas de visão computacional, como OpenCV, sem aprender todos os conceitos de CV em profundidade, como formatos de arquivo, gerenciamento de buffer, espaços de cores, valores próprios, profundidade de bits, array armazenamento, armazenamento de bitmap etc. SimpleCV alia experimentos em visão computacional usando imagens ou streams de vídeo de webcams, FireWire, telefones celulares, Kinects etc. É a melhor estrutura se precisar realizar alguma prototipagem rápida. Possui suporte aos sistemas operacionais: Mac, Windows e Ubuntu Linux.

Link: <http://simplecv.org/>

SVD Selective Vehicle Detector

VPDN Virtual Private Device Network

VPN% Velocidade Percentual N%

STAKEHOLDER Parte interessada ou partes interessadas (no plural)

Tensorflow TensorFlow é uma plataforma de código aberto gratuita que tem uma ampla variedade de ferramentas, bibliotecas e recursos para Inteligência Artificial e Aprendizado de Máquina, que inclui Visão Computacional e foi criado pela equipe do Google Brain e lançado inicialmente em 9 de novembro de 2015. É possível construir e treinar modelos de aprendizado de máquina relacionados à visão computacional que incluem reconhecimento facial, identificação de objetos etc. O Google também lançou o Pixel Visual Core (PVC) em 2017, que é um processador de imagem, visão e inteligência artificial para dispositivos móveis e é compatível com TensorFlow para aprendizado de máquina. O TensorFlow oferece suporte a linguagens como Python, C, C++, Java, JavaScript, Go, Swift etc., mas sem uma garantia de compatibilidade com versões anteriores da API. Existem também pacotes de terceiros para linguagens como MATLAB, C#, Julia, Scala, R, Rust etc.

Link: <https://www.tensorflow.org/install>

Matlab Matlab é um ambiente de computação numérica que foi desenvolvido pela MathWorks em 1984, que contém a Computer Vision Toolbox que fornece vários algoritmos e funções para visão computacional. Isso inclui detecção de objetos, rastreamento de objetos, detecção de recursos, correspondência de recursos, calibração de câmera em 3-D, reconstrução 3D etc. Pode também criar e treinar detectores de objetos personalizados usando algoritmos de aprendizado de máquina como o YOLO v2, ACF, Faster R-CNN, etc. Esses algoritmos também podem ser executados em processadores multicore e GPUs para torná-los muito mais rápidos. Os algoritmos da caixa de ferramentas do Matlab oferecem suporte à geração de código em C e C++.

Link: <https://au.mathworks.com/products/matlab.html>

YOLO YOLO ou You Only Look Once; é um sistema de detecção de objetos em tempo real mais recente e considerado de ponta; que foi criado por Joseph Redmon e Ali Farhadi da Universidade de Washington e é extremamente rápido e preciso em comparação com os outros detectores de objetos. O algoritmo YOLO é tão rápido quando comparado a outros algoritmos de detecção de objetos porque aplica

uma rede neural à imagem completa para classificar os objetos. A rede neural então particiona a imagem em regiões e prevê probabilidades para cada região. Por outro lado, o resto dos algoritmos de detecção de objetos comumente usados aplicam a rede neural a uma imagem em muitos locais e escalas diferentes. Portanto, o YOLO é rápido, pois olha para a imagem inteira, de modo que suas previsões são informadas por um contexto holístico da imagem.

Link: <https://pjreddie.com/darknet/yolo/>

Automóvel ¹	Veículo automotor destinado ao transporte de passageiros, com capacidade para até oito pessoas, exclusive o condutor.
Bonde ¹	Veículo de propulsão elétrica que se move sobre trilhos.
Caminhão ¹	Veículo automotor destinado ao transporte de carga, com carroçaria, e peso bruto total superior a 3500 Kg
Caminhão Trator ¹	Veículo automotor destinado a tracionar ou arrastar outro.
Caminhonete ¹	Veículo destinado ao transporte de carga com peso bruto total de até três mil e quinhentos quilogramas.
Camioneta ¹	Veículo misto destinado ao transporte de passageiros e carga no mesmo compartimento.
Chassi Plataforma ¹	Veículo inacabado, com equipamento que permita seu deslocamento em vias de rolamento, preparado para receber carroçaria de ônibus
Ciclomotor ¹	Veículo de duas ou três rodas, provido de um motor de combustão interna, cuja cilindrada não exceda a cinquenta centímetros cúbicos (3,05 polegadas cúbicas) e cuja velocidade máxima de fabricação não exceda a cinquenta quilômetros por hora.

¹ As definições dos veículos estão conforme o Anexo I da Lei nº 9.503, de 23 de setembro de 1997. Fonte: (BRASIL - MINISTÉRIO DA INFRAESTRUTURA, 2021)

Microônibus ¹	Veículo automotor de transporte coletivo com capacidade para até 20 passageiros.
Motocicleta ¹	Veículo automotor de duas rodas, com ou sem sidecar, dirigido por condutor em posição montada.
Motoneta ¹	Veículo automotor de duas rodas, dirigido por condutor em posição sentada.
Ônibus ¹	Veículo automotor de transporte coletivo com capacidade para mais de vinte passageiros, ainda que, em virtude de adaptações com vista à maior comodidade destes, transporte número menor.
Quadriciclo ¹	Veículo de estrutura mecânica igual às motocicletas, possuindo eixos dianteiro e traseiro, dotados de quatro rodas.
Reboque ¹	Veículo destinado a ser engatado atrás de um veículo automotor.
Semi-Reboque ¹	Veículo de um ou mais eixos que se apoia na sua unidade tratora ou é a ela ligado por meio de articulação.
Sidecar ¹	Carro ou caçamba providos de uma roda acoplada na lateral da motocicleta.
Outros ¹	Argumento que não se enquadra em nenhuma definição estabelecida.
Trator Esteira ¹	Trator que se movimenta por meio de esteira.
Trator Rodas ¹	Trator que se movimenta sobre rodas, podendo ter chassi rígido ou articulado.
Triciclo ¹	Veículo rodoviário automotor de estrutura mecânica igual à motocicleta dotado de três rodas.

Utilitário ¹	Veículo misto caracterizado pela versatilidade do seu uso, inclusive fora de estrada.
Ano de Fabricação ¹	Ano em que o veículo foi fabricado.
Marca e Modelo do veículo ¹	Nome da marca, modelo e versão do veículo.
Cor ¹	Nome da cor predominante do veículo.
Tipo de Restrição ¹	Armazena as informações de restrição do veículo.
Eixo ¹	Quantidade de eixos do veículo.
Potência ¹	Potência de um veículo expressa em CV (cavalos a vapor).
Combustível ¹	Identifica o combustível utilizado por um veículo.
Quantidade de Veículos ¹	Veículos registrados na base nacional através do sistema RENAVAM (Registro Nacional de Veículos Automotores).

APÊNDICE A – Código fonte: Obtenção dos frames de vídeos

Programa em Python, utilizado para realização dos testes:

Quadro 18 – Código completo do programa de extração de frames de arquivo de vídeo

```
# Importando todas as bibliotecas necessárias
import cv2
import os
# Leia o vídeo do caminho especificado
cam = cv2.VideoCapture("Traffic_4.mp4")
try:
    # Criando uma pasta chamada 'data'
    if not os.path.exists('data'):
        os.makedirs('data')
# Se não for criado, gera erro
except OSError:
    print('Erro: Criando diretório de dados')
# Quadro
currentframe = 0
while True:
    # Lendo FRAMES
    ret, frame = cam.read()
    if ret:
        # Se ainda sobrar vídeo, continue criando arquivos de imagens
        name = './data/frame' + str(currentframe) + '.jpg'
        print('Criando...' + name)
        # Escrevendo as imagens extraídas
        cv2.imwrite(name, frame)
        # contador crescente para mostrar quantos FRAMES são criados
        currentframe += 1
    else:
        break
# Libere o espaço e janelas quando terminado
cam.release()
cv2.destroyAllWindows()
```

Fonte: Código fonte adaptado a partir do disponibilizado por (GRANATYR, VARGAS e , 2022)

APÊNDICE B – Código fonte: Testes de desempenho dos algoritmos BGS

Programa em Python, utilizado para realização dos testes de desempenho dos algoritmos:

Quadro 19 – Código do programa de testes dos algoritmos testados

```

Contador de Veículos
# Separa em dois grupos: "Carros" e "Caminhões"
# Separa por direção de deslocamento "para Cima" e "para Baixo"
# Grava essas informações em arquivo CSV
# Grava cada veículo rastreado numa imagem do respectivo frame
# Grava vídeo com as BB inclusas

import numpy as np
import cv2
import sys
import matplotlib.pyplot as plt
import time
import csv
import validator
from random import randint

# Verificar a versão do opencv
major = cv2.__version__.split('.')[0]

# Função abaixo é para salvar os registros de cada detecção em arquivo CSV
# https://realpython.com/python-csv/#writing-csv-files-with-csv
# https://docs.python.org/pt-br/3/library/csv.html
fp = open('resultados/Comparativo 05/relatorios/relatorio.csv', mode='w')
writer_CSV = csv.DictWriter(fp, fieldnames=['Tempo', 'Veiculo'])
writer_CSV.writeheader()

# Variáveis de controle e padronização

LINE_IN_COLOR = (64, 255, 0)
LINE_OUT_COLOR = (0, 0, 255)
BOUNDING_BOX_COLOR = (255, 128, 0)
TRACKER_COLOR = (randint(0, 255), randint(0, 255), randint(0, 255))
CENTROID_COLOR = (randint(0, 255), randint(0, 255), randint(0, 255))
TEXT_COLOR = (randint(0, 255), randint(0, 255), randint(0, 255))
TEXT_POSITION_BGS = (10, 50)
TEXT_POSITION_COUNT_CARS = (10, 100)
TEXT_POSITION_COUNT_TRUCKS = (10, 150)
TEXT_SIZE = 1.2
FONT = cv2.FONT_HERSHEY_SIMPLEX
SAVE_IMAGE = True
IMAGE_DIR = "resultados/Comparativo 05/veiculos_coletados/"
VIDEO_SOURCE = "videos/Traffic_3.mp4"
VIDEO_OUT = "resultados/Comparativo 05/videos/resultado_01.avi"

# BGS - Background Subtractors
BGS_TYPES = ["GMG", "MOG", "MOG2", "KNN", "CNT", "LSBP", "GSOC"]

# Definir qual BGS usar, inserindo o número que corresponde ao BGS
selecionado no vetor de background subtractors
# 0 = GMG, 1 = MOG, 2 = MOG2, 3 = KNN, 4 = CNT, 5 = LSBP, 6 = GSOC
BGS_TYPE = BGS_TYPES[6]

```

```

# Kernel: Elemento estrutural
def getKernel(KERNEL_TYPE):
    if KERNEL_TYPE == "dilatação":
        kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (2, 2))
    if KERNEL_TYPE == "abertura":
        kernel = np.ones((3, 3), np.uint8)
    if KERNEL_TYPE == "fechamento":
        kernel = np.ones((11, 11), np.uint8)

    return kernel

# Filtros morfológicos para redução do ruído
def getFilter(img, filter):
    """
    Esses filtros são escolhidos a dedo, apenas com base em testes
    visuais
    """
    if filter == 'closing':
        return cv2.morphologyEx(img, cv2.MORPH_CLOSE,
            getKernel("fechamento"), iterations=2)

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

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

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

        return dilation

def getBGSubtractor(BGS_TYPE):
    if BGS_TYPE == "GMG":
        return
        cv2.bgsegm.createBackgroundSubtractorGMG(initializationFrames=120,
            decisionThreshold=.8)
    if BGS_TYPE == "MOG":
        return cv2.bgsegm.createBackgroundSubtractorMOG(history=200,
            nmixtures=5, backgroundRatio=.7, noiseSigma=0)
    if BGS_TYPE == "MOG2":
        return cv2.createBackgroundSubtractorMOG2(history=50,
            detectShadows=False, varThreshold=200)
    if BGS_TYPE == "KNN":
        return cv2.createBackgroundSubtractorKNN(history=100,
            dist2Threshold=400, detectShadows=True)
    if BGS_TYPE == "CNT":
        return
        cv2.bgsegm.createBackgroundSubtractorCNT(minPixelStability=15,
            useHistory=True,

```

```

maxPixelStability=15 * 60, isParallel=True)
    if BGS_TYPE == "LSBP":
        return cv2.bgsegm.createBackgroundSubtractorLSBP()
    if BGS_TYPE == "GSOC":
        return cv2.bgsegm.createBackgroundSubtractorGSOC()
    print("Tipo de BGS desconhecido")
    sys.exit(1)

# Calcular o centro de cada objeto
def getCentroid(x, y, w, h):
    x1 = int(w / 2)
    y1 = int(h / 2)

    cx = x + x1
    cy = y + y1

    return (cx, cy)

def save_frame(frame, file_name, flip=True):
    # Filtro para Mudar de RGB para BGR
    if flip:
        cv2.imwrite(file_name, np.flip(frame, 2))
    else:
        cv2.imwrite(file_name, frame)

# Carregar o video
cap = cv2.VideoCapture(VIDEO_SOURCE)
hasFrame, frame = cap.read()
# Grava o video
fourcc = cv2.VideoWriter_fourcc(*'XVID')
writer_VIDEO = cv2.VideoWriter(VIDEO_OUT, fourcc, 25, (frame.shape[1],
frame.shape[0]), True)

# Limites da ROI (R.egion O.f I.interest)
# Definimos uma região de interesse para diminuir a interferência de outros
objetos na imagem.
# Por exemplo, contando veículos e uma pessoa passar pela lateral da pista,
o algoritmo irá perceber o movimento.
# Agora se focar somente no movimento da pista, onde os veículos passam,
ignorando o que está fora da ROI não teremos este problema
# h: altura, w: largura
# h1, h2 = 100, 300
# w1, w2 = 350, 900
bbox = cv2.selectROI(frame, False)
(w1, h1, w2, h2) = bbox
# largura, altura, largura, altura
print(bbox)

# Limiar das dimensões do objeto na ROI (-->> valor empírico)
frameArea = h2 * w2
minArea = int(frameArea / 250)
maxArea = 8000
print('Limiar de Área', minArea)

```

```

# Linhas limites de entrada/saída (line_DOWN, line_UP)
# Para que um objeto seja contado, terá que entrar pela área de entrada,
cruzar a linha de entrada (line_DOWN) e sair pela área de saída, cruzando a
linha de saída (line_UP)
line_DOWN = int(h1 + 20)

# Linha de saída está posiciona a 20 pixels a menos na vertical, ou seja,
ela é mais alta que o limite da ROI
# h2 = 300, então 300-20 = 280 na vertical
line_UP = int(h2 - 20)

# Limiar de entrada do objeto na ROI na coordenada Y
# O objeto precisa estar dentro deste limiar (--> valor empírico)
UP_limit = int(h1 / 5)
DOWN_limit = int(h1 / 4)

print("Limite SUPERIOR y:", str(UP_limit))
print("Limite INFERIOR y:", str(DOWN_limit))

bg_subtractor = getBGSubtractor(BGS_TYPE)

def main():
    frame_number = -1

    # Para armazenar a contagem de objetos que cruzaram a linha
    cnt_cars, cnt_trucks = 0, 0

    # Vetor para armazenar os objetos
    objects = []

    # Tempo limite de vida/existência do objeto na classe
    max_p_age = 2

    # Inicialização do ID dos objetos
    pid = 1

    while (cap.isOpened()):

        ok, frame = cap.read()
        if not ok:
            print("A captura do quadro falhou, parando...")
            break

        # Limitar o processamento das imagens à somente a área de interesse
        (ROI)
        roi = frame[h1:h1 + h2, w1:w1 + w2]

        # Percorrer o vetor de objetos
        for i in objects:
            i.age_one() # marcar cada detecção como um objeto

        # Usamos o número do frame para servir com índice ao salvar os
        resultados
        frame_number += 1

        # Aplicar o BGS na ROI
        bg_mask = bg_subtractor.apply(roi)

```

```

# Tenta aplicar os filtros morfológicos, caso ocorra algum erro,
retorna 0 (zero) para a contagem de carros e caminhões
try:
    bg_mask = getFilter(bg_mask, 'combine')
except:
    print('CARROS:', cnt_cars)
    print('CAMINHOES:', cnt_trucks)
    break

# Se a versão do OpenCV for 3, então usa 3 parâmetros
if major == '3':
    (img, contours, hierarchy) = cv2.findContours(bg_mask,
cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    else: # Nas versões mais recentes do OpenCV, cv2.findContours()
foi alterada para retornar apenas os contornos e a hierarquia
    (contours, hierarchy) = cv2.findContours(bg_mask,
cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

# Percorrer os contornos
for cnt in contours:
    area = cv2.contourArea(cnt)

    # Contando carros
    if area > minArea and area <= maxArea:
        x, y, w, h = cv2.boundingRect(cnt)

        # Calcular o centroide dos objetos
        centroid = getCentroid(x, y, w, h)
        cx = centroid[0]
        cy = centroid[1]

        new = True

        # Desenhar um pequeno retângulo preenchido acima do objeto
e escrever qual o tipo
        cv2.rectangle(roi, (x, y), (x + 50, y - 13), TRACKER_COLOR,
-1)
        cv2.putText(roi, 'Carro', (x, y - 2), FONT, .5, (255, 255,
255), 1, cv2.LINE_AA)

# Percorrer o vetor de objetos
for i in objects:
    # O método abs() retorna o valor absoluto do número
    # Exemplo:
    # floating = -30.33
    # print('O valor absoluto de -30,33 é:', abs(floating))
    # O valor absoluto de -30,33 é: 30,33
    # Se o valor absoluto do novo objeto na horizontal (x)
menos o valor na horizontal (getX) do objeto na lista de objetos for menor
ou igual ao valor na horizontal do objeto
    # E Se o valor absoluto do novo objeto na vertical (y)
menos o valor na vertical (getY) do objeto na lista de objetos for menor ou
igual ao valor na vertical do objeto
    # Então não é um novo objeto, somente atualizamos as
coordenadas
    if abs(x - i.getX()) <= w and abs(y - i.getY()) <= h:

```

```

        new = False
        i.updateCoords(cx, cy)
        # Verificar se o objeto desceu, ou seja, indo de
cima para baixo
        if i.going_DOWN(DOWN_limit) == True:
            cnt_cars += 1
            if SAVE_IMAGE:
                save_frame(roi, IMAGE_DIR +
"/Carro_Descendo_%04d.png" % frame_number,
                            flip=False) # Salvar a imagem
de cada veículo
                writer_CSV.writerow({'Tempo':
time.strftime("%c"), 'Veiculo': 'Carro Descendo ' + str(
                            cnt_cars)}) # Gravar o resultado em
CSV
                print("ID:", i.getId(), 'atravessou indo para
baixo em', time.strftime("%c"))

        # Verificar se o objeto subiu, ou seja, indo de
baixo para cima
        elif i.going_UP(UP_limit) == True:
            cnt_cars += 1
            if SAVE_IMAGE:
                save_frame(roi, IMAGE_DIR +
"/Carro_Subindo_%04d.png" % frame_number,
                            flip=False) # Salvar a imagem
de cada veículo
                writer_CSV.writerow({'Tempo':
time.strftime("%c"), 'Veiculo': 'Carro Subindo ' + str(
                            cnt_cars)}) # Gravar o resultado em
CSV
                print("ID:", i.getId(), 'cruzou subindo em',
time.strftime("%c"))
            break
        if i.getState() == '1':
            # Verificar se o objeto cruzou a linha de saída,
vindo de cima para baixo
            if i.getDir() == 'Descendo' and i.getY() > line_UP:
                i.setDone()
            # Verificar se o objeto cruzou a linha de saída,
vindo de baixo para cima
            if i.getDir() == 'Subindo' and i.getY() <
line_DOWN:
                i.setDone()
        if i.timeOut():
            index = objects.index(i)
            objects.pop(index)
            del i
        if new == True:
            p = validator.MyValidator(pid, cx, cy, max_p_age)
            objects.append(p)
            pid += 1

        cv2.circle(roi, (cx, cy), 5, CENTROID_COLOR, -1)
        img = cv2.rectangle(roi, (x, y), (x + w, y + h),
TRACKER_COLOR, 2)

        # Contando caminhos
        elif area >= maxArea:
            x, y, w, h = cv2.boundingRect(cnt)

```

```

centroid = getCentroid(x, y, w, h)
cx = centroid[0]
cy = centroid[1]

new = True

cv2.rectangle(roi, (x, y), (x + 50, y - 13), TRACKER_COLOR,
-1)
cv2.putText(roi, 'Caminhao', (x, y - 2), FONT, .5, (255,
255, 255), 1, cv2.LINE_AA)

# Percorrer o vetor de objetos
for i in objects:
    if abs(x - i.getX()) <= w and abs(y - i.getY()) <= h:
        new = False
        i.updateCoords(cx, cy)

        # Verificar se o objeto desceu, ou seja, indo de
cima para baixo
        if i.going_DOWN(DOWN_limit) == True:
            cnt_trucks += 1
            if SAVE_IMAGE:
                save_frame(roi, IMAGE_DIR +
"/Caminhao_Descendo_%04d.png" % frame_number, flip=False)
                writer_CSV.writerow(
                    {'Tempo': time.strftime("%c"),
'Veiculo': 'Caminhao Descendo ' + str(cnt_trucks)})
                print("ID:", i.getId(), 'atravessou indo para
baixo em', time.strftime("%c"))

            # Verificar se o objeto subiu, ou seja, indo de
baixo para cima
            elif i.going_UP(UP_limit) == True:
                cnt_trucks += 1
                if SAVE_IMAGE:
                    save_frame(roi, IMAGE_DIR +
"/Caminhao_Subindo_%04d.png" % frame_number,
                                flip=False) # Salvar a imagem
de cada veículo
                    writer_CSV.writerow({'Tempo':
time.strftime("%c"), 'Veiculo': 'Caminhao Subindo ' + str(
                                cnt_trucks)}) # Gravar o resultado em
CSV
                    print("ID:", i.getId(), 'cruzou subindo em',
time.strftime("%c"))
                break
            if i.getState() == '1':
                # Verificar se o objeto cruzou a linha de saída,
vindo de cima para baixo
                if i.getDir() == 'Descendo' and i.getY() > line_UP:
                    i.setDone()
                # Verificar se o objeto cruzou a linha de saída,
vindo de baixo para cima
                if i.getDir() == 'Subindo' and i.getY() <
line_DOWN:
                    i.setDone()
            if i.timedOut():
                index = objects.index(i)

```

```

        objects.pop(index)
        del i
    if new == True:
        p = validator.MyValidator(pid, cx, cy, max_p_age)
        objects.append(p)
        pid += 1

    cv2.circle(roi, (cx, cy), 5, CENTROID_COLOR, -1)
    img = cv2.rectangle(roi, (x, y), (x + w, y + h),
TRACKER_COLOR, 2)

    for i in objects:
        cv2.putText(roi, str(i.getId()), (i.getX(), i.getY()),
            FONT, 0.3, TEXT_COLOR, 1, cv2.LINE_AA)

    str_cars = 'Carros: ' + str(cnt_cars)
    str_trucks = 'Caminhoes: ' + str(cnt_trucks)

    frame = cv2.line(frame, (w1, line_DOWN), (w1 + w2, line_DOWN),
LINE_IN_COLOR, 2)
    frame = cv2.line(frame, (w1, h1 + line_UP), (w1 + w2, h1 +
line_UP), LINE_OUT_COLOR, 2)

    cv2.putText(frame, 'Background Subtractor: ' + BGS_TYPE,
TEXT_POSITION_BGS, FONT, TEXT_SIZE, (255, 255, 255), 4,
        cv2.LINE_AA)
    cv2.putText(frame, 'Background Subtractor: ' + BGS_TYPE,
TEXT_POSITION_BGS, FONT, TEXT_SIZE, (128, 0, 255), 2,
        cv2.LINE_AA)

    cv2.putText(frame, str_cars, TEXT_POSITION_COUNT_CARS, FONT,
TEXT_SIZE, (255, 255, 255), 4, cv2.LINE_AA)
    cv2.putText(frame, str_cars, TEXT_POSITION_COUNT_CARS, FONT,
TEXT_SIZE, (255, 128, 0), 2, cv2.LINE_AA)

    cv2.putText(frame, str_trucks, TEXT_POSITION_COUNT_TRUCKS, FONT,
TEXT_SIZE, (255, 255, 255), 4, cv2.LINE_AA)
    cv2.putText(frame, str_trucks, TEXT_POSITION_COUNT_TRUCKS, FONT,
TEXT_SIZE, (255, 128, 0), 2, cv2.LINE_AA)

    # Sobreposições transparentes
    # https://www.pyimagesearch.com/2016/03/07/transparent-overlays-
with-opencv/
    for alpha in np.arange(0.3, 1.1, 0.9)[::-1]:
        overlay = frame.copy()
        output = frame.copy()
        cv2.rectangle(overlay, (w1, h1), (w1 + w2, h1 + h2),
BOUNDING_BOX_COLOR, -1)
        frame = cv2.addWeighted(overlay, alpha, output, 1 - alpha, 0,
output)

    # Função para visualizar o modelo fundo
    # Disponível somente para MOG2, KNN, CNT
    if BGS_TYPE != 'MOG' and BGS_TYPE != 'GMG':
        bg = bg_subtractor.getBackgroundImage()
        # cv2.imshow('Model', bg)
    cv2.imshow('Frame', frame)
    cv2.imshow('ROI_Mask', bg_mask)

```

```
        if SAVE_IMAGE:
            writer_VIDEO.write(frame)

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

writer_VIDEO.release()
cap.release()
cv2.destroyAllWindows()

main()
```

Fonte: Código fonte adaptado a partir do disponibilizado por (GRANATYR, VARGAS e , 2022)

Módulo necessário - Programa: validator.py

Quadro 20 – Código do programa (módulo obrigatório) – validator.py

```
import math

class MyValidator:
    tracks = []

    def __init__(self, i, xi, yi, max_age):
        self.i = i
        self.x = xi
        self.y = yi
        self.tracks = []
        self.done = False
        self.state = '0'
        self.age = 0
        self.max_age = max_age
        self.dir = None

    def getTracks(self):
        return self.tracks

    def getId(self):
        return self.i

    def getState(self):
        return self.state

    def getDir(self):
        return self.dir

    def getX(self):
        return self.x

    def getY(self):
        return self.y

    def updateCoords(self, xn, yn):
        self.age = 0
        self.tracks.append([self.x, self.y])
        self.x = xn
        self.y = yn

    def setDone(self):
        self.done = True

    def timedOut(self):
        return self.done

    def going_DOWN(self, mid_start):

        # Verificar se tem pelo menos 2 coordenadas de objetos armazenadas
        if len(self.tracks) >= 2:
```

```

# Verificar se o estado do objeto é zero
# O estado do objeto só mudará quando cruzar o limiar de
entrada
if self.state == '0':
    # Cálculo da distância euclidiana
    distance = math.sqrt(float((self.tracks[-1][1] -
self.tracks[-2][1])**2) + float(
    (self.tracks[-1][1] - self.tracks[-2][1])**2))
    if distance < 10:
        # [-2] são duas posições anteriores do registro no
vetor e [1] é a coluna contendo os
        # valores na vertical (y) de cada objeto
        # Se a posição vertical anterior do objeto for maior
que o limiar de entrada e se em
        # duas posições verticais anteriores o valor for
menor ou igual ao limiar de entrada
        # Então atualizamos o estado do objeto para 1 e
indicamos que o mesmo se moveu na
        # direção para baixo (down)
        # Fazemos isso para ter certeza de que o objeto cruzou
a linha de entrada, movendo-se
        # de cima para baixo
        if self.tracks[-1][1] > mid_start and self.tracks[-
2][1] <= mid_start:
            state = '1'
            self.dir = 'down'
            return True
        else:
            return False
    else:
        return False
def going_UP(self, mid_end):
    if len(self.tracks) >= 2:
        if self.state == '0':
            distance = math.sqrt(float((self.tracks[-1][1] -
self.tracks[-2][1])**2) + float(
            (self.tracks[-1][1] - self.tracks[-2][1])**2))
            if distance < 10:
                if self.tracks[-1][1] < mid_end and self.tracks[-2][1]
>= mid_end:
                    state = '1'
                    self.dir = 'up'
                    return True
                else:
                    return False
            else:
                return False
def age_one(self):
    self.age += 1
    if self.age > self.max_age:
        self.done = True
    return True

```

Fonte: Código fonte adaptado a partir do disponibilizado por (GRANATYR, VARGAS e , 2022)

APÊNDICE C – Equipamentos utilizados

Lista do equipamento e softwares utilizados.

- Notebook ACER ASPIRE

Especificações do dispositivo:

Processador Intel Core i7-6500U CPU 2.50GHz

Memória RAM instalada: 16GB

HD principal: 480GB SSD Kingston SA400S37480G

HD secundário: 2TB Seagate ST2000LM003 HN-M201RAD

Video primário: Intel HD Graphics 520

Video secundário: Nvidia Geforce 940M

Sistema Operacional: Windows 10 Pro 64bits

Versão: 22H2

Compilação: 19045.2311

APÊNDICE D – Controle - vídeo Traffic_3.mp4

Imagens dos frames extraídos do vídeo original Traffic_3.mp4 para controle, apenas constam as imagens com os registros dos veículos contabilizados de forma manual, não representando desta forma a totalidade dos frames obtidos.

Tabela 20 - Controle - Contagem manual dos veículos do vídeo Traffic_3.mp4

ID	Arquivo de Imagem	QTD Carros	QTD Caminhões	QTD outros
1	Frame0	0	0	0
2	Frame20	1	0	0
3	frame80	3	0	0
4	frame133	2	0	0
5	frame193	1	0	0
6	frame262	4	0	0
7	frame307	4	0	0
8	frame356	3	0	0
9	frame374	1	1	0
10	frame408	3	0	0
11	frame490	2	1	0
12	frame527	2	0	0
13	frame616	2	0	0
14	frame673	3	0	0
15	frame710	1	0	0
16	frame790	1	0	0
17	frame838	3	0	0
18	frame933	3	0	0
19	frame950	1	0	0
20	frame1098	3	0	0
21	frame1130	0	1	0
22	frame1165	2	1	0
23	frame1190	1	0	0
24	frame1235	4	0	0
25	frame1264	2	0	0
26	frame1305	1	1	0
27	frame1458	1	0	0
28	frame1478	1	0	0
29	frame1570	1	1	0
30	frame1678	2	0	0
31	frame1705	1	0	0
32	frame1768	3	0	0
33	frame1851	1	0	0
34	frame1900	1	0	0
35	frame1920	2	0	0
36	frame1950	2	0	0
37	frame1996	1	0	0
38	frame2044	3	0	0
39	frame2084	1	1	0

Continua

Continuação

ID	Arquivo de Imagem	QTD Carros	QTD Caminhões	QTD outros
40	frame2113	1	0	0
41	frame2210	3	0	0
42	frame2265	1	0	0
43	frame2321	3	0	0
44	frame2351	1	0	0
45	frame2407	1	0	0
46	frame2470	2	1	0
47	frame2517	2	2	0
48	frame2560	3	0	0
49	frame2620	3	0	0
50	frame2654	1	1	0
51	frame2718	3	0	0
52	frame2750	3	1	0
53	frame2873	4	1	0
54	frame2905	1	0	0
55	frame2943	0	1	0
56	frame3020	1	1	0
57	frame3118	1	0	0
58	frame3162	2	0	0
59	frame3233	1	1	0
60	frame3278	1	0	0
61	frame3316	4	0	0
62	frame3338	1	0	0
63	frame3380	3	0	0
64	frame3440	3	1	0
65	frame3490	1	0	0
66	frame3575	1	0	0
67	frame3660	3	0	0
68	frame3727	2	0	0
69	frame3819	2	0	0
70	frame3851	2	0	0
71	frame3892	3	1	0
72	frame3924	4	0	0
73	frame3950	4	0	0
74	frame3998	6	0	0
75	frame4080	3	0	0
76	frame4177	4	0	0
77	frame4188	1	0	0
78	frame4255	2	0	0
79	frame4314	4	1	0

Continua

Continuação

ID	Arquivo de Imagem	QTD Carros	QTD Caminhões	QTD outros
80	frame4352	1	0	0
81	frame4455	3	1	0
82	frame4480	1	0	0
83	frame4563	2	0	0
84	frame4688	3	0	0
85	frame4730	3	0	0
86	frame4790	1	0	0
87	frame4832	3	2	0
88	frame4848	1	0	0
89	frame4871	1	0	0
90	frame4968	2	0	0
91	frame4982	1	0	0
92	frame5245	1	0	0
93	frame5292	0	1	0
94	frame5415	3	1	0
95	frame5525	4	0	0
96	frame5572	4	0	0
97	frame5603	3	0	0
98	frame5672	2	0	0
99	frame5735	1	0	0
100	frame5790	2	0	0
101	frame5822	2	0	0
102	frame5850	1	0	0
103	frame5965	2	1	0
104	frame6090	4	0	0
105	frame6121	3	0	0
106	frame6187	1	0	0
107	frame6257	3	0	0
108	frame6295	2	0	0
109	frame6370	1	0	0
110	frame6410	3	1	0
111	frame6438	1	0	0
112	frame6480	4	0	0
113	frame6533	1	0	0
114	frame6585	2	0	0
115	frame6615	2	0	0
116	frame6665	1	0	0
117	frame6727	1	0	0
118	frame6768	4	0	0
119	frame6805	2	0	0

Continua

Continuação

ID	Arquivo de Imagem	QTD Carros	QTD Caminhões	QTD outros
120	frame6832	3	0	0
121	frame6910	1	0	0
122	frame6936	1	0	0
123	frame7030	2	0	0
124	frame7151	3	0	0
125	frame7254	2	0	0
126	frame7330	3	1	0
127	frame7360	4	0	0
128	frame7400	2	0	0
129	frame7413	1	0	0
130	frame7485	2	0	0
131	frame7556	1	1	0
132	frame7574	1	0	0
133	frame7641	3	0	0
134	frame7696	4	0	0
135	frame7777	4	0	0
136	frame7835	2	0	0
137	frame7896	3	0	0
138	frame7960	1	0	0
139	frame8135	3	0	0
140	frame8157	3	0	0
141	frame8224	5	0	0
142	frame8257	1	0	0
143	frame8390	3	0	0
144	frame8553	2	0	0
145	frame8650	3	0	0
146	frame8700	1	0	0
147	frame8815	1	0	0
148	frame8870	1	0	0
149	frame8940	1	0	0
150	frame9045	2	0	0
151	frame9100	0	1	0
152	frame9140	1	0	0
153	frame9180	2	0	0
154	frame9225	3	0	0
155	frame9285	4	0	0
156	frame9346	2	0	0
157	frame9435	3	0	0
158	frame9480	1	0	0
159	frame9570	3	0	0

Continua

Continuação

ID	Arquivo de Imagem	QTD Carros	QTD Caminhões	QTD outros
160	frame9750	6	0	0
161	frame9816	2	0	0
162	frame9857	1	0	0
163	frame9906	1	0	0
164	frame9930	1	0	0
165	frame10010	3	0	0
166	frame10047	2	0	0
167	frame10080	2	2	0
168	frame10123	3	0	0
169	frame10180	1	0	0
170	frame10288	3	0	0
171	frame10337	2	0	0
172	frame10390	1	0	0
173	frame10547	2	0	0
174	frame10590	1	0	0
175	frame10614	1	0	0
176	frame10675	3	0	0
177	frame10730	0	1	0
178	frame10820	2	0	0
179	frame10855	1	0	0
180	frame10980	3	0	0
181	frame11038	3	1	0
182	frame11090	1	0	0
183	frame11160	2	0	0
184	frame11190	2	0	0
185	frame11230	2	0	0
186	frame11300	0	1	0
187	frame11339	1	0	0
188	frame11430	2	1	0
189	frame11450	1	0	0
190	frame11481	1	1	0
191	frame11510	1	0	0
192	frame11600	2	0	0
193	frame11630	1	0	0
194	frame11700	2	0	0
195	frame11720	1	0	0
196	frame11770	3	0	0
197	frame11915	2	0	0
198	frame11964	4	0	0
199	frame12023	2	0	0

Continua

Continuação

ID	Arquivo de Imagem	QTD Carros	QTD Caminhões	QTD outros
200	frame12070	3	0	0
201	frame12115	3	0	0
202	frame12230	2	0	0
203	frame12285	0	1	0
204	frame12360	2	1	0
205	frame12450	1	1	0
206	frame12490	1	0	0
207	frame12528	1	0	0
208	frame12560	1	0	0
209	frame12607	2	0	0
210	frame12655	3	0	0
211	frame12685	1	0	0
212	frame12800	2	0	0
213	frame12850	1	0	0
214	frame12860	2	0	0
215	frame12894	1	0	0
216	frame12920	2	0	0
217	frame12970	0	0	1 moto
218	frame12980	2	0	0
219	frame13040	1	0	0
220	frame13148	2	0	0
221	frame13196	1	0	0
222	frame13246	1	0	0
223	frame13365	1	0	0
224	frame13416	1	0	0
225	frame13445	2	0	0
226	frame13464	1	0	0
227	frame13500	1	0	0
228	frame13540	3	0	0
229	frame13642	3	0	0
230	frame13663	2	0	0
231	frame13695	1	0	0
232	frame13740	1	0	0
233	frame13752	1	0	0
234	frame13840	1	0	0
235	frame13910	1	0	0
236	frame13948	3	1	0
237	frame13964	1	0	0
238	frame14106	4	0	0
239	frame14165	2	0	0

Continua

Continuação

ID	Arquivo de Imagem	QTD Carros	QTD Caminhões	QTD outros
240	frame14226	1	1	0
241	frame14273	2	0	0
242	frame14452	1	0	0
243	frame14614	2	0	0
244	frame14648	2	0	0
245	frame14695	2	0	0
246	frame14720	2	0	0
247	frame14754	1	0	0
248	frame14866	2	0	0
249	frame14920	3	0	0
250	frame14952	1	0	0
251	frame14972	1	1	0
252	frame15016	2	0	0
253	frame15066	1	0	0
254	frame15104	2	0	0
255	frame15134	2	0	0
256	frame15168	2	1	0
257	frame15208	1	0	0
258	frame15280	2	0	0
259	frame15316	2	0	0
260	frame15386	3	0	0
261	frame15408	1	0	0
262	frame15488	1	0	0
263	frame15560	1	0	0
264	frame15690	2	1	0
265	frame15808	3	0	0
266	frame15910	2	0	0
267	frame15926	2	0	0
268	frame15952	1	0	0
269	frame16036	1	0	0
270	frame16136	1	0	0
271	frame16248	1	0	0
272	frame16338	3	0	0
273	frame16354	2	0	0
274	frame16394	1	0	0
275	frame16420	3	0	0
276	frame16436	1	0	0
277	frame16460	2	0	0
278	frame16530	0	1	0
279	frame16628	1	0	0

Continua

Continuação

ID	Arquivo de Imagem	QTD Carros	QTD Caminhões	QTD outros
280	frame16694	1	1	0
281	frame16758	2	0	0
282	frame16832	2	0	0
283	frame16852	2	0	0
284	frame16914	3	0	0
285	frame16944	2	0	0
286	frame16990	2	0	0
287	frame17028	1	1	0
288	frame17076	2	0	0
289	frame17182	2	0	0
290	frame17202	1	1	0
291	frame17250	1	0	0
292	frame17282	3	0	0
293	frame17324	2	1	0
294	frame17340	1	0	0
295	frame17354	2	0	0
296	frame17398	1	0	0
297	frame17420	3	0	0
298	frame17436	3	0	0
299	frame17470	2	0	0
300	frame17522	2	0	0
301	frame17546	2	1	0
302	frame17572	1	0	0
303	frame17628	1	1	0
304	frame17650	2	0	0
305	frame17678	4	0	0
306	frame17788	2	0	0
307	frame17834	3	0	0
308	frame17916	5	0	0
309	frame18024	2	0	0
310	frame18112	3	0	0
311	frame18210	1	0	0
312	frame18224	1	0	0
313	frame18272	1	1	0
314	frame18312	1	0	0
315	frame18376	2	1	0
316	frame18422	2	0	0
317	frame18495	2	1	0
318	frame18538	1	0	0
319	frame18590	2	0	0

Continua

Continuação

ID	Arquivo de Imagem	QTD Carros	QTD Caminhões	QTD outros
320	frame18674	2	0	0
321	frame18714	4	0	0
322	frame18762	3	0	0
323	frame18818	2	0	0
324	frame18846	1	0	0
325	frame18870	1	0	0
326	frame18904	4	0	0
327	frame18930	1	0	0
328	frame18992	1	0	0
329	frame19054	2	0	0
330	frame19092	1	1	0
331	frame19184	3	0	0
332	frame19296	3	0	0
333	frame19326	1	0	0
334	frame19338	1	0	0
335	frame19350	1	0	0
336	frame19370	3	0	0
337	frame19432	3	0	0
338	frame19444	1	0	0
339	frame19476	1	1	0
340	frame19502	1	0	0
341	frame19536	2	1	0
342	frame19560	1	0	0
343	frame19632	3	0	0
344	frame19732	2	0	0
345	frame19764	1	0	0
346	frame19786	2	1	0
347	frame19828	1	0	0
348	frame19842	1	0	0
349	frame19892	3	0	0
350	frame19910	1	0	0
351	frame19984	3	0	0
352	frame20048	1	0	0
353	frame20072	1	0	0
354	frame20162	2	0	0
355	frame20208	2	1	0
356	frame20312	2	1	0
357	frame20334	1	0	0
358	frame20360	1	0	0
359	frame20408	2	0	0

Continua

Continuação

ID	Arquivo de Imagem	QTD Carros	QTD Caminhões	QTD outros
360	frame20460	1	0	0
361	frame20528	1	1	0
362	frame20548	1	0	0
363	frame20570	3	0	0
364	frame20614	1	0	0
365	frame20640	1	0	0
366	frame20660	1	0	0
367	frame20734	3	1	0
368	frame20747	1	0	0
369	frame20784	1	0	0
370	frame20806	2	0	0
371	frame20842	2	0	0
372	frame20890	1	0	0
373	frame21002	2	0	0
374	frame21080	1	0	0
375	frame21136	2	0	0
376	frame21212	2	0	0
377	frame21264	1	0	0
378	frame21312	2	0	0
379	frame21366	4	0	0
380	frame21488	3	0	0
381	frame21514	1	0	0
382	frame21528	1	0	0
383	frame21622	1	1	0
384	frame21674	4	0	0
385	frame21792	1	0	0
386	frame21856	2	0	0
387	frame21908	1	0	0
388	frame21932	2	0	0
389	frame21966	1	1	0
390	frame21982	2	0	0
391	frame22026	1	1	0
392	frame22080	1	0	0
393	frame22118	1	0	0
394	frame22172	2	0	0
395	frame22272	6	0	0
396	frame22302	2	0	0
397	frame22336	1	0	0
398	frame22356	3	0	0
399	frame22390	2	0	0

Continua

Continuação

ID	Arquivo de Imagem	QTD Carros	QTD Caminhões	QTD outros
400	frame22472	2	0	0
401	frame22526	1	1	0
402	frame22622	3	0	0
403	frame22734	3	0	0
404	frame22778	1	1	0
405	frame22830	2	0	0
406	frame22866	1	0	0
407	frame22962	2	0	0
408	frame23070	1	0	0
409	frame23178	2	0	0
410	frame23220	4	0	0
411	frame23236	2	0	0
412	frame23284	1	0	0
413	frame23330	2	1	0
414	frame23362	4	0	0
415	frame23398	3	0	0
416	frame23434	3	0	0
417	frame23502	1	0	0
418	frame23530	3	0	0
419	frame23604	4	0	0
420	frame23650	2	0	0
421	frame23668	0	0	1 moto
422	frame23770	2	0	0
423	frame23864	5	0	0
424	frame23938	3	0	0
425	frame24078	4	0	0
426	frame24150	3	0	0
427	frame24172	0	0	1 moto
428	frame24212	2	1	0
429	frame24238	1	0	0
430	frame24272	1	1	0
431	frame24334	2	0	0
432	frame24362	2	0	0
433	frame24394	2	0	0
434	frame24430	1	0	0
435	frame24454	2	0	0
436	frame24500	4	0	0
437	frame24544	2	0	0
438	frame24600	5	0	0
439	frame24632	1	0	0

Continua

Continuação

ID	Arquivo de Imagem	QTD Carros	QTD Caminhões	QTD outros
440	frame24676	1	0	0
441	frame24708	1	0	0
442	frame24768	1	0	0
443	frame24824	3	1	0
444	frame24852	2	0	0
445	frame24878	1	0	0
446	frame24928	2	1	0
447	frame24970	2	0	0
448	frame25110	4	0	0
449	frame25138	2	0	0
450	frame25178	1	0	0
451	frame25189	1	0	0

Fonte: o autor

APÊNDICE E – Frames extraídos do vídeo Traffic_3.mp4

Imagens dos frames extraídos do vídeo original Traffic_3.mp4 para controle, apenas constam as imagens com os registros dos veículos contabilizados de forma manual, não representando desta forma a totalidade dos frames obtidos.

Imagem 7 – Frame 0 - Status inicial do vídeo Traffic_3.mp4



Fonte: O autor

Quadro 21 - Frames extraídos para o controle da contagem manual dos veículos do vídeo Traffic_3.mp4

Imagem 8 – Frame 20 | Traffic_3.mp4



Imagem 9 – Frame 80 | Traffic_3.mp4



Imagem 10 – Frame 133 | Traffic_3.mp4



Imagem 11 - Frame 193 | Traffic_3.mp4



Imagem 12 - Frame 262 | Traffic_3.mp4



Imagem 13 - Frame 307 | Traffic_3.mp4



Imagem 14 - Frame 356 | Traffic_3.mp4



Imagem 15 - Frame 374 | Traffic_3.mp4



Continua

Continuação

Imagem 16 - Frame 408 | Traffic_3.mp4



Imagem 17 - Frame 673 | Traffic_3.mp4

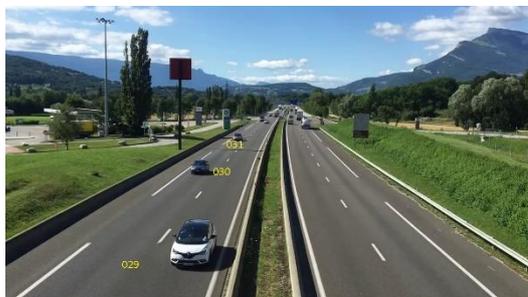


Imagem 18 - Frame 490 | Traffic_3.mp4



Imagem 19 - Frame 710 | Traffic_3.mp4



Imagem 20 - Frame 527 | Traffic_3.mp4



Imagem 21 - Frame 790 | Traffic_3.mp4



Imagem 22 - Frame 616 | Traffic_3.mp4



Imagem 23 - Frame 838 | Traffic_3.mp4



Continua

Imagem 24 - Frame 933 | Traffic_3.mp4

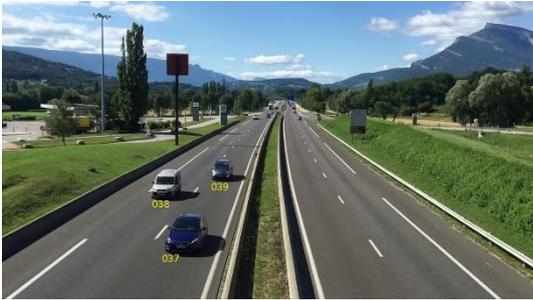


Imagem 25 - Frame 950 | Traffic_3.mp4



Imagem 26 - Frame 1098 | Traffic_3.mp4



Imagem 27 - Frame 1130 | Traffic_3.mp4



Imagem 28 - Frame 1165 | Traffic_3.mp4



Imagem 29 - Frame 1190 | Traffic_3.mp4



Imagem 30 - Frame 1235 | Traffic_3.mp4



Imagem 31 - Frame 1264 | Traffic_3.mp4



Continua

Continuação

Imagem 32 - Frame 1305 | Traffic_3.mp4



Imagem 33 - Frame 1458 | Traffic_3.mp4



Imagem 34 - Frame 1478 | Traffic_3.mp4



Imagem 35 - Frame 1570 | Traffic_3.mp4



Imagem 36 - Frame 1678 | Traffic_3.mp4



Imagem 37 - Frame 1705 | Traffic_3.mp4



Imagem 38 - Frame 1768 | Traffic_3.mp4



Imagem 39 - Frame 1851 | Traffic_3.mp4



Continua

Continuação

Imagem 40 - Frame 1900 | Traffic_3.mp4



Imagem 41 - Frame 1920 | Traffic_3.mp4



Imagem 42 - Frame 1950 | Traffic_3.mp4



Imagem 43 - Frame 1996 | Traffic_3.mp4



Imagem 44 - Frame 2044 | Traffic_3.mp4

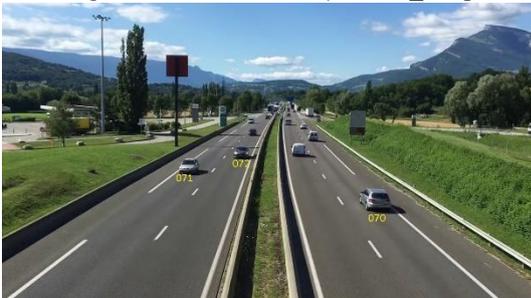


Imagem 45 - Frame 2084 | Traffic_3.mp4



Imagem 46 - Frame 2113 | Traffic_3.mp4



Imagem 47 - Frame 2210 | Traffic_3.mp4



Continua

Continuação

Imagem 48 - Frame 2265 | Traffic_3.mp4



Imagem 49 - Frame 2321 | Traffic_3.mp4



Imagem 50 - Frame 2351 | Traffic_3.mp4



Imagem 51 - Frame 2407 | Traffic_3.mp4



Imagem 52 - Frame 2470 | Traffic_3.mp4



Imagem 53 - Frame 2517 | Traffic_3.mp4



Imagem 54 - Frame 2560 | Traffic_3.mp4



Imagem 55 - Frame 2620 | Traffic_3.mp4



Continua

Continuação

Imagem 56 - Frame 2654 | Traffic_3.mp4



Imagem 57 - Frame 2718 | Traffic_3.mp4



Imagem 58 - Frame 2750 | Traffic_3.mp4



Imagem 59 - Frame 2873 | Traffic_3.mp4



Imagem 60 - Frame 2905 | Traffic_3.mp4



Imagem 61 - Frame 2943 | Traffic_3.mp4



Imagem 62 - Frame 3020 | Traffic_3.mp4



Imagem 63 - Frame 3118 | Traffic_3.mp4



Continua

Continuação

Imagem 64 - Frame 3162 | Traffic_3.mp4



Imagem 65 - Frame 3233 | Traffic_3.mp4



Imagem 66 - Frame 3278 | Traffic_3.mp4



Imagem 67 - Frame 3316 | Traffic_3.mp4



Imagem 68 - Frame 3338 | Traffic_3.mp4



Imagem 69 - Frame 3380 | Traffic_3.mp4



Imagem 70 - Frame 3440 | Traffic_3.mp4



Imagem 71 - Frame 3490 | Traffic_3.mp4



Continua

Continuação

Imagem 72 - Frame 3575 | Traffic_3.mp4



Imagem 73 - Frame 3660 | Traffic_3.mp4



Imagem 74 - Frame 3727 | Traffic_3.mp4



Imagem 75 - Frame 3819 | Traffic_3.mp4



Imagem 76 - Frame 3851 | Traffic_3.mp4



Imagem 77 - Frame 3892 | Traffic_3.mp4



Imagem 78 - Frame 3924 | Traffic_3.mp4



Imagem 79 - Frame 3950 | Traffic_3.mp4



Continua

Continuação

Imagem 80 - Frame 3998 | Traffic_3.mp4

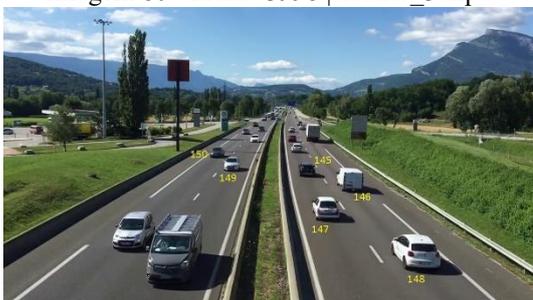


Imagem 81 - Frame 4080 | Traffic_3.mp4



Imagem 82 - Frame 4177 | Traffic_3.mp4



Imagem 83 - Frame 4188 | Traffic_3.mp4



Imagem 84 - Frame 4255 | Traffic_3.mp4



Imagem 85 - Frame 4314 | Traffic_3.mp4



Imagem 86 - Frame 4352 | Traffic_3.mp4



Imagem 87 - Frame 4455 | Traffic_3.mp4



Continua

Continuação

Imagem 88 - Frame 4480 | Traffic_3.mp4



Imagem 89 - Frame 4563 | Traffic_3.mp4



Imagem 90 - Frame 4688 | Traffic_3.mp4



Imagem 91 - Frame 4730 | Traffic_3.mp4



Imagem 92 - Frame 4790 | Traffic_3.mp4



Imagem 93 - Frame 4832 | Traffic_3.mp4



Imagem 94 - Frame 4848 | Traffic_3.mp4



Imagem 95 - Frame 4871 | Traffic_3.mp4



Continua

Continuação

Imagem 96 - Frame 4968 | Traffic_3.mp4



Imagem 97 - Frame 4982 | Traffic_3.mp4



Imagem 98 - Frame 5245 | Traffic_3.mp4



Imagem 99 - Frame 5292 | Traffic_3.mp4



Imagem 100 - Frame 5415 | Traffic_3.mp4



Imagem 101 - Frame 5525 | Traffic_3.mp4



Imagem 102 - Frame 5572 | Traffic_3.mp4



Imagem 103 - Frame 5603 | Traffic_3.mp4



Continua

Continuação

Imagem 104 - Frame 5672 | Traffic_3.mp4



Imagem 105 - Frame 5735 | Traffic_3.mp4



Imagem 106 - Frame 5790 | Traffic_3.mp4



Imagem 107 - Frame 5822 | Traffic_3.mp4



Imagem 108 - Frame 5850 | Traffic_3.mp4



Imagem 109 - Frame 5965 | Traffic_3.mp4



Imagem 110 - Frame 6090 | Traffic_3.mp4

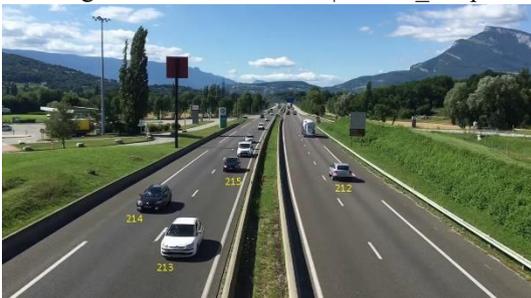


Imagem 111 - Frame 6121 | Traffic_3.mp4



Continua

Continuação

Imagem 112 - Frame 6187 | Traffic_3.mp4



Imagem 113 - Frame 6257 | Traffic_3.mp4



Imagem 114 - Frame 6295 | Traffic_3.mp4



Imagem 115 - Frame 6370 | Traffic_3.mp4



Imagem 116 - Frame 6410 | Traffic_3.mp4



Imagem 117 - Frame 6438 | Traffic_3.mp4



Imagem 118 - Frame 6480 | Traffic_3.mp4



Imagem 119 - Frame 6533 | Traffic_3.mp4



Continua

Continuação

Imagem 120 - Frame 6585 | Traffic_3.mp4



Imagem 121 - Frame 6615 | Traffic_3.mp4



Imagem 122 - Frame 6665 | Traffic_3.mp4



Imagem 123 - Frame 6727 | Traffic_3.mp4



Imagem 124 - Frame 6768 | Traffic_3.mp4



Imagem 125 - Frame 6805 | Traffic_3.mp4



Imagem 126 - Frame 6832 | Traffic_3.mp4



Imagem 127 - Frame 6910 | Traffic_3.mp4



Continua

Continuação

Imagem 128 - Frame 6936 | Traffic_3.mp4



Imagem 129 - Frame 7030 | Traffic_3.mp4



Imagem 130 - Frame 7151 | Traffic_3.mp4

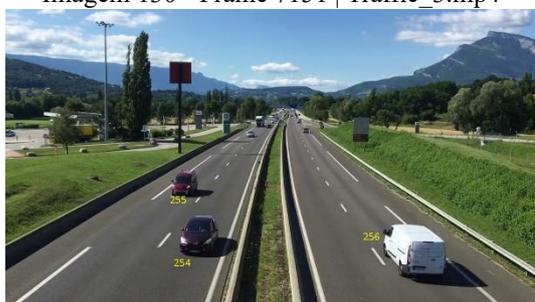


Imagem 131 - Frame 7254 | Traffic_3.mp4



Imagem 132 - Frame 7330 | Traffic_3.mp4



Imagem 133 - Frame 7360 | Traffic_3.mp4



Imagem 134 - Frame 7400 | Traffic_3.mp4



Imagem 135 - Frame 7413 | Traffic_3.mp4



Continua

Continuação

Imagem 136 - Frame 7485 | Traffic_3.mp4



Imagem 137 - Frame 7556 | Traffic_3.mp4



Imagem 138 - Frame 7574 | Traffic_3.mp4



Imagem 139 - Frame 7641 | Traffic_3.mp4



Imagem 140 - Frame 7696 | Traffic_3.mp4



Imagem 141 - Frame 7777 | Traffic_3.mp4



Imagem 142 - Frame 7835 | Traffic_3.mp4



Imagem 143 - Frame 7896 | Traffic_3.mp4



Continua

Continuação

Imagem 144 - Frame 7960 | Traffic_3.mp4



Imagem 145 - Frame 8135 | Traffic_3.mp4



Imagem 146 - Frame 8157 | Traffic_3.mp4



Imagem 147 - Frame 8224 | Traffic_3.mp4



Imagem 148 - Frame 8257 | Traffic_3.mp4



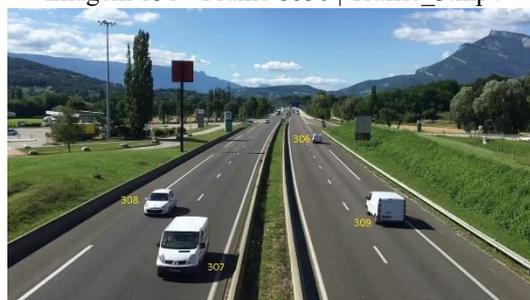
Imagem 149 - Frame 8390 | Traffic_3.mp4



Imagem 150 - Frame 8553 | Traffic_3.mp4



Imagem 151 - Frame 8650 | Traffic_3.mp4



Continua

Continuação

Imagem 152 - Frame 8700 | Traffic_3.mp4



Imagem 153 - Frame 8815 | Traffic_3.mp4



Imagem 154 - Frame 8870 | Traffic_3.mp4



Imagem 155 - Frame 8940 | Traffic_3.mp4



Imagem 156 - Frame 9045 | Traffic_3.mp4



Imagem 157 - Frame 9100 | Traffic_3.mp4



Imagem 158 - Frame 9140 | Traffic_3.mp4



Imagem 159 - Frame 9180 | Traffic_3.mp4



Continua

Continuação

Imagem 160 - Frame 9225 | Traffic_3.mp4



Imagem 161 - Frame 9285 | Traffic_3.mp4



Imagem 162 - Frame 9346 | Traffic_3.mp4



Imagem 163 - Frame 9435 | Traffic_3.mp4



Imagem 164 - Frame 9480 | Traffic_3.mp4



Imagem 165 - Frame 9570 | Traffic_3.mp4



Imagem 166 - Frame 9750 | Traffic_3.mp4



Imagem 167 - Frame 9816 | Traffic_3.mp4



Continua

Continuação

Imagem 168 - Frame 9857 | Traffic_3.mp4



Imagem 169 - Frame 9906 | Traffic_3.mp4



Imagem 170 - Frame 9930 | Traffic_3.mp4



Imagem 171 - Frame 10010 | Traffic_3.mp4



Imagem 172 - Frame 10047 | Traffic_3.mp4



Imagem 173 - Frame 10080 | Traffic_3.mp4



Imagem 174 - Frame 10123 | Traffic_3.mp4



Imagem 175 - Frame 10180 | Traffic_3.mp4



Continua

Continuação

Imagem 176 - Frame 10288 | Traffic_3.mp4



Imagem 177 - Frame 10337 | Traffic_3.mp4



Imagem 178 - Frame 10390 | Traffic_3.mp4



Imagem 179 - Frame 10547 | Traffic_3.mp4



Imagem 180 - Frame 10590 | Traffic_3.mp4



Imagem 181 - Frame 10614 | Traffic_3.mp4



Imagem 182 - Frame 10675 | Traffic_3.mp4



Imagem 183 - Frame 10730 | Traffic_3.mp4



Continua

Continuação

Imagem 184 - Frame 10820 | Traffic_3.mp4

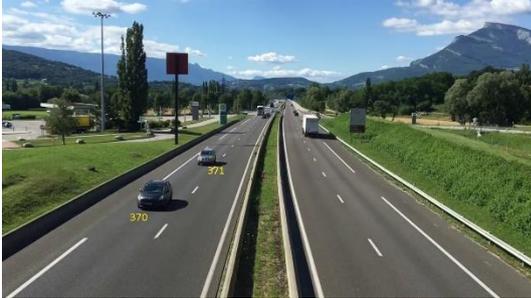


Imagem 185 - Frame 10855 | Traffic_3.mp4



Imagem 186 - Frame 10980 | Traffic_3.mp4



Imagem 187 - Frame 11038 | Traffic_3.mp4



Imagem 188 - Frame 11090 | Traffic_3.mp4



Imagem 189 - Frame 11160 | Traffic_3.mp4



Imagem 190 - Frame 11190 | Traffic_3.mp4



Imagem 191 - Frame 11230 | Traffic_3.mp4



Continua

Continuação

Imagem 192 - Frame 11300 | Traffic_3.mp4



Imagem 193 - Frame 11339 | Traffic_3.mp4



Imagem 194 - Frame 11430 | Traffic_3.mp4



Imagem 195 - Frame 11450 | Traffic_3.mp4



Imagem 196 - Frame 11481 | Traffic_3.mp4



Imagem 197 - Frame 11510 | Traffic_3.mp4



Imagem 198 - Frame 11600 | Traffic_3.mp4



Imagem 199 - Frame 11630 | Traffic_3.mp4



Continua

Continuação

Imagem 200 - Frame 11700 | Traffic_3.mp4



Imagem 201 - Frame 11720 | Traffic_3.mp4



Imagem 202 - Frame 11770 | Traffic_3.mp4



Imagem 203 - Frame 11915 | Traffic_3.mp4



Imagem 204 - Frame 11964 | Traffic_3.mp4



Imagem 205 - Frame 12023 | Traffic_3.mp4



Imagem 206 - Frame 12070 | Traffic_3.mp4



Imagem 207 - Frame 12115 | Traffic_3.mp4



Continua

Continuação

Imagem 208 - Frame 12230 | Traffic_3.mp4



Imagem 209 - Frame 12285 | Traffic_3.mp4



Imagem 210 - Frame 12360 | Traffic_3.mp4



Imagem 211 - Frame 12450 | Traffic_3.mp4



Imagem 212 - Frame 12490 | Traffic_3.mp4



Imagem 213 - Frame 12528 | Traffic_3.mp4



Imagem 214 - Frame 12560 | Traffic_3.mp4



Imagem 215 - Frame 12607 | Traffic_3.mp4



Continua

Continuação

Imagem 216 - Frame 12655 | Traffic_3.mp4



Imagem 217 - Frame 12685 | Traffic_3.mp4



Imagem 218 - Frame 12800 | Traffic_3.mp4



Imagem 219 - Frame 12850 | Traffic_3.mp4



Imagem 220 - Frame 12860 | Traffic_3.mp4



Imagem 221 - Frame 12894 | Traffic_3.mp4



Imagem 222 - Frame 12920 | Traffic_3.mp4



Imagem 223 - Frame 12970 | Traffic_3.mp4



Continua

Continuação

Imagem 224 - Frame 12980 | Traffic_3.mp4



Imagem 225 - Frame 13040 | Traffic_3.mp4



Imagem 226 - Frame 13148 | Traffic_3.mp4



Imagem 227 - Frame 13196 | Traffic_3.mp4



Imagem 228 - Frame 13246 | Traffic_3.mp4



Imagem 229 - Frame 13365 | Traffic_3.mp4



Imagem 230 - Frame 13416 | Traffic_3.mp4



Imagem 231 - Frame 13445 | Traffic_3.mp4



Continua

Continuação

Imagem 232 - Frame 13464 | Traffic_3.mp4



Imagem 233 - Frame 13500 | Traffic_3.mp4



Imagem 234 - Frame 13540 | Traffic_3.mp4

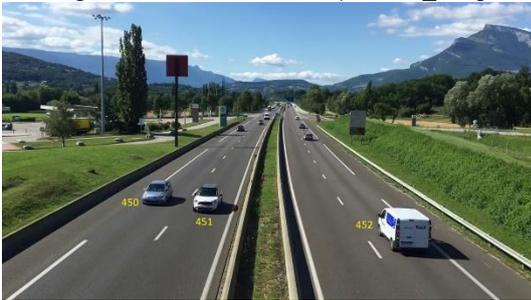


Imagem 235 - Frame 13642 | Traffic_3.mp4



Imagem 236 - Frame 13663 | Traffic_3.mp4



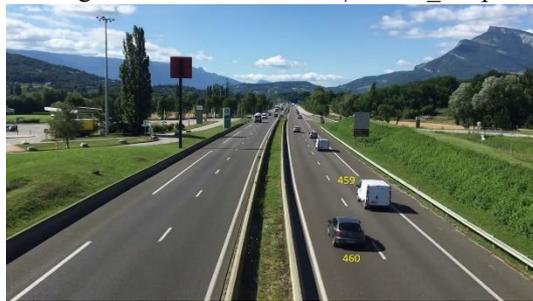
Imagem 237 - Frame 13695 | Traffic_3.mp4



Imagem 238 - Frame 13740 | Traffic_3.mp4



Imagem 239 - Frame 13752 | Traffic_3.mp4



Continua

Continuação

Imagem 240 - Frame 13840 | Traffic_3.mp4



Imagem 241 - Frame 13910 | Traffic_3.mp4



Imagem 242 - Frame 13948 | Traffic_3.mp4



Imagem 243 - Frame 13964 | Traffic_3.mp4



Imagem 244 - Frame 14106 | Traffic_3.mp4



Imagem 245 - Frame 14165 | Traffic_3.mp4



Imagem 246 - Frame 14226 | Traffic_3.mp4



Imagem 247 - Frame 14273 | Traffic_3.mp4



Continua

Continuação

Imagem 248 - Frame 14452 | Traffic_3.mp4



Imagem 249 - Frame 14614 | Traffic_3.mp4



Imagem 250 - Frame 14648 | Traffic_3.mp4

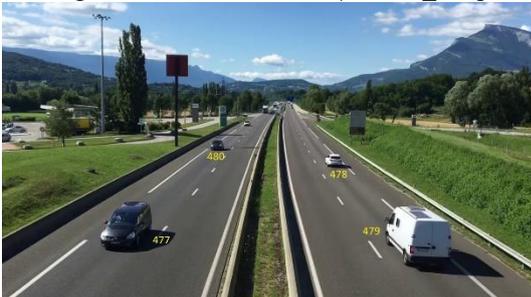


Imagem 251 - Frame 14695 | Traffic_3.mp4



Imagem 252 - Frame 14720 | Traffic_3.mp4



Imagem 253 - Frame 14754 | Traffic_3.mp4



Imagem 254 - Frame 14866 | Traffic_3.mp4



Imagem 255 - Frame 14920 | Traffic_3.mp4



Continua

Continuação

Imagem 256 - Frame 14952 | Traffic_3.mp4



Imagem 257 - Frame 14972 | Traffic_3.mp4



Imagem 258 - Frame 15016 | Traffic_3.mp4



Imagem 259 - Frame 15066 | Traffic_3.mp4



Imagem 260 - Frame 15104 | Traffic_3.mp4



Imagem 261 - Frame 15134 | Traffic_3.mp4



Imagem 262 - Frame 15168 | Traffic_3.mp4



Imagem 263 - Frame 15208 | Traffic_3.mp4



Continua

Continuação

Imagem 264 - Frame 15280 | Traffic_3.mp4

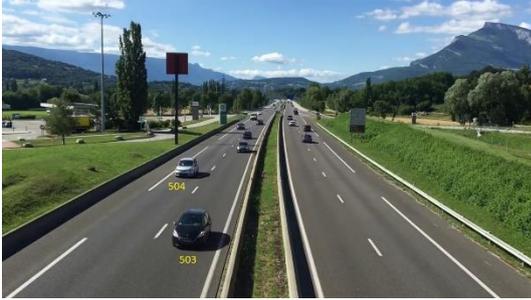


Imagem 265 - Frame 15316 | Traffic_3.mp4



Imagem 266 - Frame 15386 | Traffic_3.mp4



Imagem 267 - Frame 15408 | Traffic_3.mp4

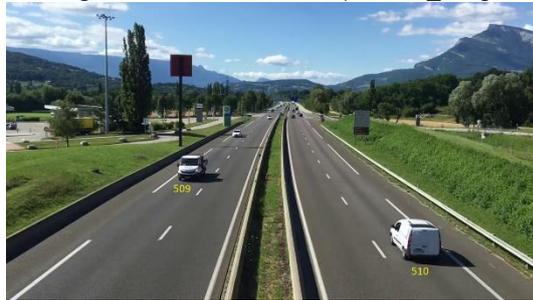


Imagem 268 - Frame 15488 | Traffic_3.mp4



Imagem 269 - Frame 15560 | Traffic_3.mp4



Imagem 270 - Frame 15690 | Traffic_3.mp4



Imagem 271 - Frame 15808 | Traffic_3.mp4



Continua

Continuação

Imagem 272 - Frame 15910 | Traffic_3.mp4



Imagem 273 - Frame 15926 | Traffic_3.mp4



Imagem 274 - Frame 15952 | Traffic_3.mp4



Imagem 275 - Frame 16036 | Traffic_3.mp4



Imagem 276 - Frame 16136 | Traffic_3.mp4



Imagem 277 - Frame 16248 | Traffic_3.mp4



Imagem 278 - Frame 16338 | Traffic_3.mp4



Imagem 279 - Frame 16354 | Traffic_3.mp4



Continua

Continuação

Imagem 280 - Frame 16394 | Traffic_3.mp4



Imagem 281 - Frame 16420 | Traffic_3.mp4



Imagem 282 - Frame 16436 | Traffic_3.mp4



Imagem 283 - Frame 16460 | Traffic_3.mp4



Imagem 284 - Frame 16530 | Traffic_3.mp4



Imagem 285 - Frame 16628 | Traffic_3.mp4



Imagem 286 - Frame 16694 | Traffic_3.mp4



Imagem 287 - Frame 16758 | Traffic_3.mp4



Continua

Continuação

Imagem 288 - Frame 16832 | Traffic_3.mp4



Imagem 289 - Frame 16852 | Traffic_3.mp4



Imagem 290 - Frame 16914 | Traffic_3.mp4



Imagem 291 - Frame 16944 | Traffic_3.mp4



Imagem 292 - Frame 16990 | Traffic_3.mp4



Imagem 293 - Frame 17028 | Traffic_3.mp4

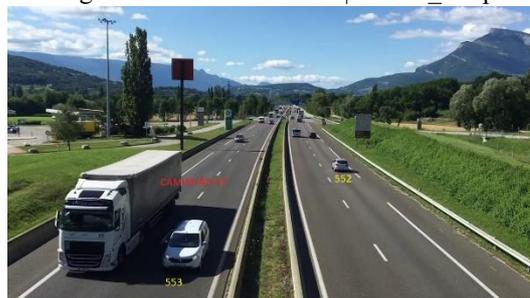


Imagem 294 - Frame 17076 | Traffic_3.mp4



Imagem 295 - Frame 17182 | Traffic_3.mp4



Continua

Continuação

Imagem 296 - Frame 17202 | Traffic_3.mp4

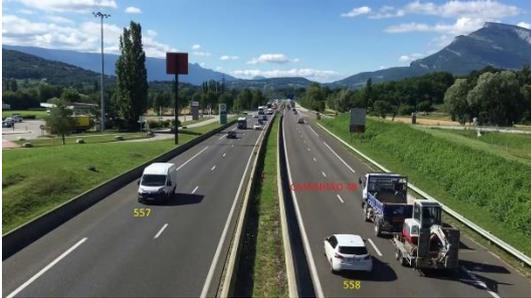


Imagem 297 - Frame 17250 | Traffic_3.mp4



Imagem 298 - Frame 17282 | Traffic_3.mp4



Imagem 299 - Frame 17324 | Traffic_3.mp4



Imagem 300 - Frame 17340 | Traffic_3.mp4



Imagem 301 - Frame 17354 | Traffic_3.mp4



Imagem 302 - Frame 17398 | Traffic_3.mp4



Imagem 303 - Frame 17420 | Traffic_3.mp4



Continua

Continuação

Imagem 304 - Frame 17436 | Traffic_3.mp4



Imagem 305 - Frame 17470 | Traffic_3.mp4



Imagem 306 - Frame 17522 | Traffic_3.mp4



Imagem 307 - Frame 17546 | Traffic_3.mp4



Imagem 308 - Frame 17572 | Traffic_3.mp4



Imagem 309 - Frame 17628 | Traffic_3.mp4



Imagem 310 - Frame 17650 | Traffic_3.mp4



Imagem 311 - Frame 17678 | Traffic_3.mp4



Continua

Continuação

Imagem 312 - Frame 17788 | Traffic_3.mp4



Imagem 313 - Frame 17834 | Traffic_3.mp4



Imagem 314 - Frame 17916 | Traffic_3.mp4



Imagem 315 - Frame 18024 | Traffic_3.mp4



Imagem 316 - Frame 18112 | Traffic_3.mp4



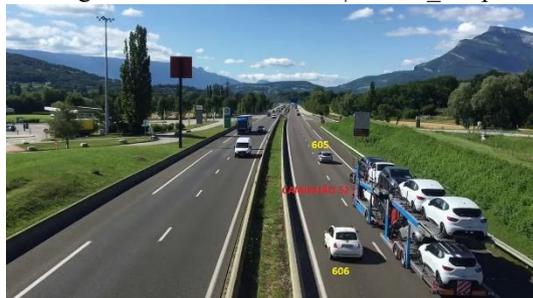
Imagem 317 - Frame 18210 | Traffic_3.mp4



Imagem 318 - Frame 18224 | Traffic_3.mp4



Imagem 319 - Frame 18272 | Traffic_3.mp4



Continua

Continuação

Imagem 320 - Frame 18312 | Traffic_3.mp4



Imagem 321 - Frame 18376 | Traffic_3.mp4



Imagem 322 - Frame 18422 | Traffic_3.mp4



Imagem 323 - Frame 18495 | Traffic_3.mp4



Imagem 324 - Frame 18538 | Traffic_3.mp4



Imagem 325 - Frame 18590 | Traffic_3.mp4



Imagem 326 - Frame 18674 | Traffic_3.mp4



Imagem 327 - Frame 18714 | Traffic_3.mp4



Continua

Continuação

Imagem 328 - Frame 18762 | Traffic_3.mp4



Imagem 329 - Frame 18818 | Traffic_3.mp4



Imagem 330 - Frame 18846 | Traffic_3.mp4

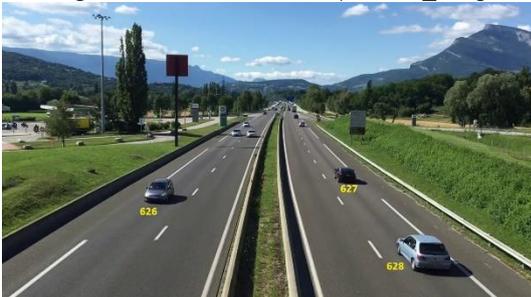


Imagem 331 - Frame 18870 | Traffic_3.mp4



Imagem 332 - Frame 18904 | Traffic_3.mp4



Imagem 333 - Frame 18930 | Traffic_3.mp4



Imagem 334 - Frame 18992 | Traffic_3.mp4



Imagem 335 - Frame 19054 | Traffic_3.mp4



Continua

Continuação

Imagem 336 - Frame 19092 | Traffic_3.mp4



Imagem 337 - Frame 19184 | Traffic_3.mp4



Imagem 338 - Frame 19296 | Traffic_3.mp4



Imagem 339 - Frame 19326 | Traffic_3.mp4



Imagem 340 - Frame 19338 | Traffic_3.mp4



Imagem 341 - Frame 19350 | Traffic_3.mp4



Imagem 342 - Frame 19370 | Traffic_3.mp4



Imagem 343 - Frame 19432 | Traffic_3.mp4



Continua

Continuação

Imagem 344 - Frame 19444 | Traffic_3.mp4



Imagem 345 - Frame 19476 | Traffic_3.mp4



Imagem 346 - Frame 19502 | Traffic_3.mp4



Imagem 347 - Frame 19536 | Traffic_3.mp4

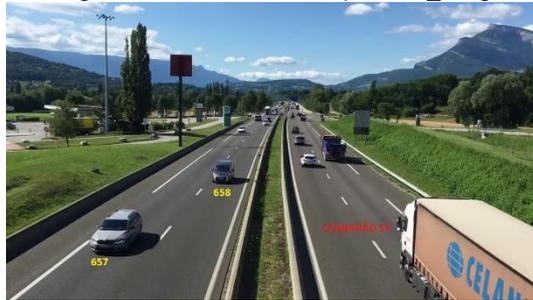


Imagem 348 - Frame 19560 | Traffic_3.mp4



Imagem 349 - Frame 19632 | Traffic_3.mp4



Imagem 350 - Frame 19732 | Traffic_3.mp4



Imagem 351 - Frame 19764 | Traffic_3.mp4



Continua

Continuação

Imagem 352 - Frame 19786 | Traffic_3.mp4



Imagem 353 - Frame 19828 | Traffic_3.mp4



Imagem 354 - Frame 19842 | Traffic_3.mp4



Imagem 355 - Frame 19892 | Traffic_3.mp4



Imagem 356 - Frame 19910 | Traffic_3.mp4



Imagem 357 - Frame 19984 | Traffic_3.mp4

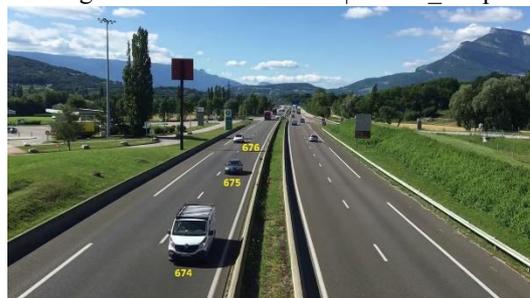


Imagem 358 - Frame 20048 | Traffic_3.mp4



Imagem 359 - Frame 20072 | Traffic_3.mp4



Continua

Continuação

Imagem 360 - Frame 20162 | Traffic_3.mp4



Imagem 361 - Frame 20208 | Traffic_3.mp4



Imagem 362 - Frame 20312 | Traffic_3.mp4

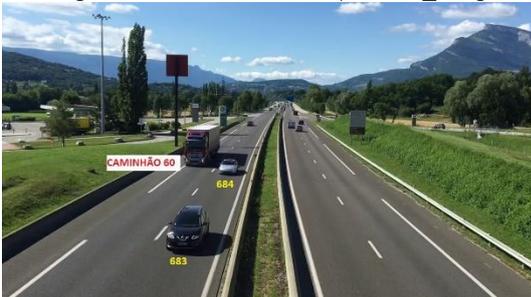


Imagem 363 - Frame 20334 | Traffic_3.mp4

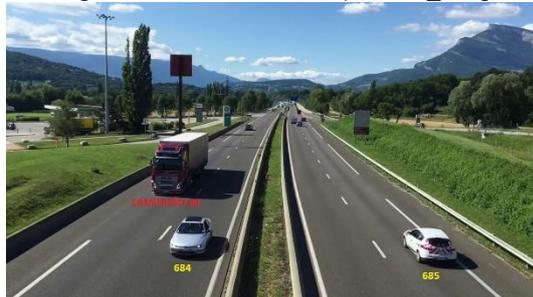


Imagem 364 - Frame 20360 | Traffic_3.mp4



Imagem 365 - Frame 20408 | Traffic_3.mp4



Imagem 366 - Frame 20460 | Traffic_3.mp4



Imagem 367 - Frame 20528 | Traffic_3.mp4



Continua

Continuação

Imagem 368 - Frame 20548 | Traffic_3.mp4



Imagem 369 - Frame 20570 | Traffic_3.mp4

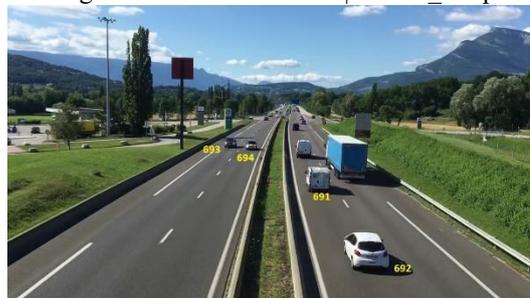


Imagem 370 - Frame 20614 | Traffic_3.mp4



Imagem 371 - Frame 20640 | Traffic_3.mp4



Imagem 372 - Frame 20660 | Traffic_3.mp4



Imagem 373 - Frame 20734 | Traffic_3.mp4

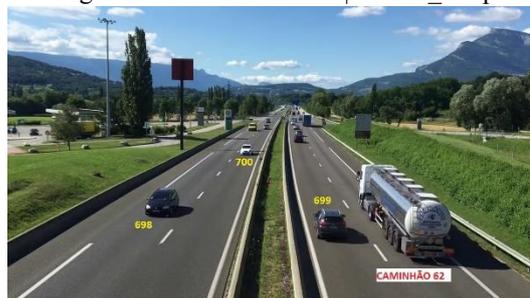


Imagem 374 - Frame 20747 | Traffic_3.mp4



Imagem 375 - Frame 20784 | Traffic_3.mp4



Continua

Continuação

Imagem 376 - Frame 20806 | Traffic_3.mp4



Imagem 377 - Frame 20842 | Traffic_3.mp4

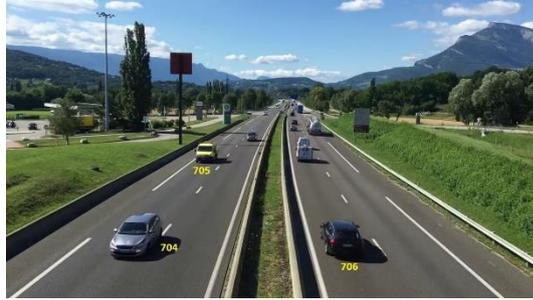


Imagem 378 - Frame 20890 | Traffic_3.mp4



Imagem 379 - Frame 21002 | Traffic_3.mp4

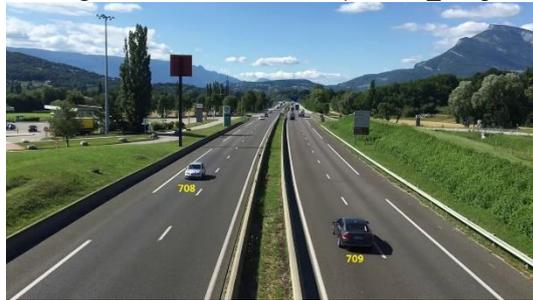


Imagem 380 - Frame 21080 | Traffic_3.mp4



Imagem 381 - Frame 21136 | Traffic_3.mp4

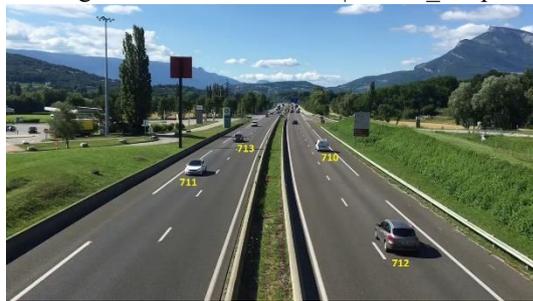
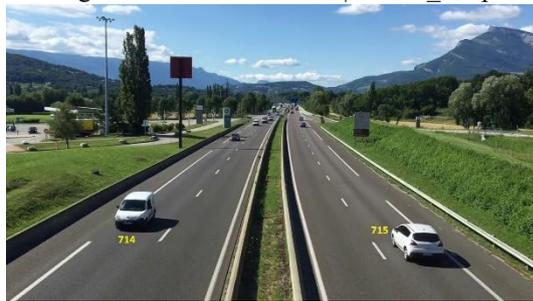


Imagem 382 - Frame 21212 | Traffic_3.mp4



Imagem 383 - Frame 21264 | Traffic_3.mp4



Continua

Continuação

Imagem 384 - Frame 21312 | Traffic_3.mp4



Imagem 385 - Frame 21366 | Traffic_3.mp4



Imagem 386 - Frame 21488 | Traffic_3.mp4



Imagem 387 - Frame 21514 | Traffic_3.mp4



Imagem 388 - Frame 21528 | Traffic_3.mp4



Imagem 389 - Frame 21622 | Traffic_3.mp4



Imagem 390 - Frame 21674 | Traffic_3.mp4



Imagem 391 - Frame 21792 | Traffic_3.mp4



Continua

Continuação

Imagem 392 - Frame 21856 | Traffic_3.mp4



Imagem 393 - Frame 21908 | Traffic_3.mp4



Imagem 394 - Frame 21932 | Traffic_3.mp4



Imagem 395 - Frame 21966 | Traffic_3.mp4



Imagem 396 - Frame 21982 | Traffic_3.mp4

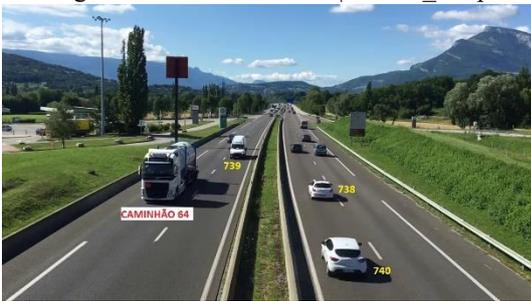


Imagem 397 - Frame 22026 | Traffic_3.mp4



Imagem 398 - Frame 22080 | Traffic_3.mp4



Imagem 399 - Frame 22118 | Traffic_3.mp4



Continua

Continuação

Imagem 400 - Frame 22172 | Traffic_3.mp4



Imagem 401 - Frame 22272 | Traffic_3.mp4

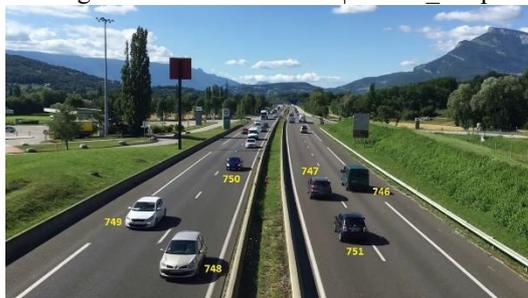


Imagem 402 - Frame 22302 | Traffic_3.mp4



Imagem 403 - Frame 22336 | Traffic_3.mp4



Imagem 404 - Frame 22356 | Traffic_3.mp4

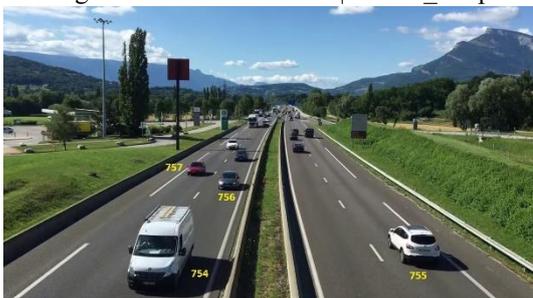


Imagem 405 - Frame 22390 | Traffic_3.mp4



Imagem 406 - Frame 22472 | Traffic_3.mp4



Imagem 407 - Frame 22526 | Traffic_3.mp4



Continua

Continuação

Imagem 408 - Frame 22622 | Traffic_3.mp4



Imagem 409 - Frame 22734 | Traffic_3.mp4



Imagem 410 - Frame 22778 | Traffic_3.mp4



Imagem 411 - Frame 22830 | Traffic_3.mp4



Imagem 412 - Frame 22866 | Traffic_3.mp4



Imagem 413 - Frame 22962 | Traffic_3.mp4



Imagem 414 - Frame 23070 | Traffic_3.mp4



Imagem 415 - Frame 23178 | Traffic_3.mp4



Continua

Continuação

Imagem 416 - Frame 23220 | Traffic_3.mp4



Imagem 417 - Frame 23236 | Traffic_3.mp4



Imagem 418 - Frame 23284 | Traffic_3.mp4



Imagem 419 - Frame 23330 | Traffic_3.mp4



Imagem 420 - Frame 23362 | Traffic_3.mp4



Imagem 421 - Frame 23398 | Traffic_3.mp4

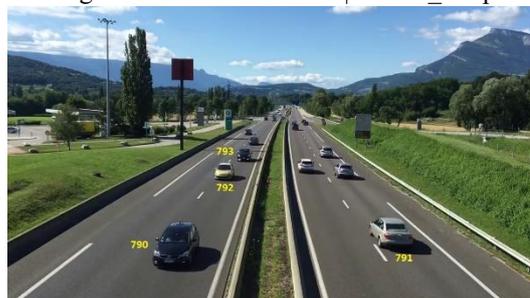


Imagem 422 - Frame 23434 | Traffic_3.mp4



Imagem 423 - Frame 23502 | Traffic_3.mp4



Continua

Continuação

Imagem 424 - Frame 23530 | Traffic_3.mp4

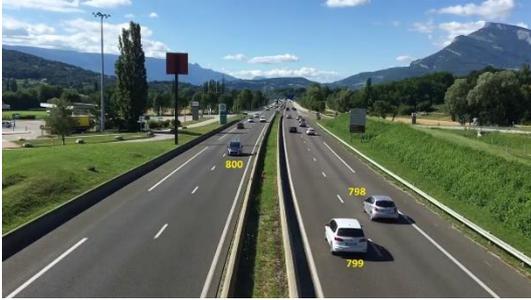


Imagem 425 - Frame 23604 | Traffic_3.mp4

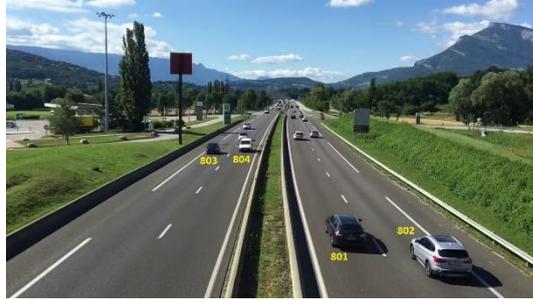


Imagem 426 - Frame 23650 | Traffic_3.mp4

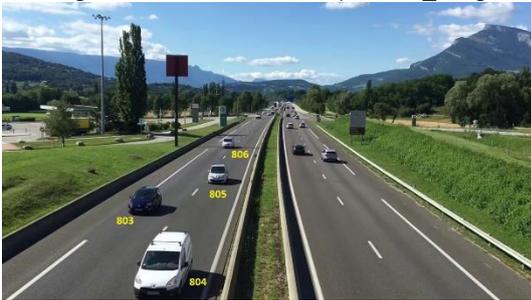


Imagem 427 - Frame 23668 | Traffic_3.mp4

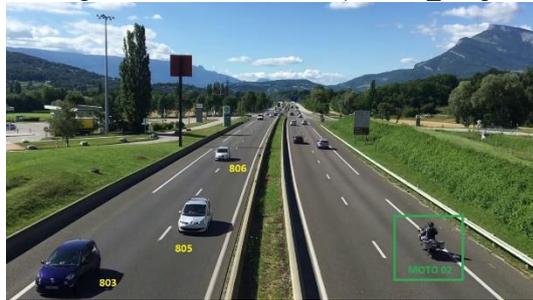


Imagem 428 - Frame 23770 | Traffic_3.mp4



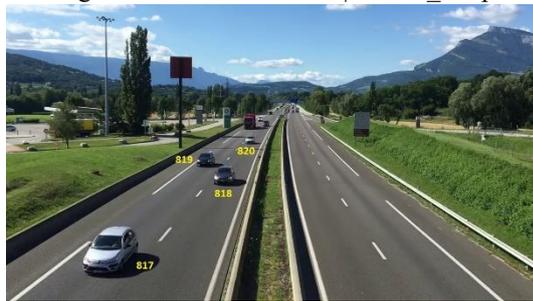
Imagem 429 - Frame 23864 | Traffic_3.mp4



Imagem 430 - Frame 23938 | Traffic_3.mp4



Imagem 431 - Frame 24078 | Traffic_3.mp4



Continua

Continuação

Imagem 432 - Frame 24150 | Traffic_3.mp4



Imagem 433 - Frame 24172 | Traffic_3.mp4



Imagem 434 - Frame 24212 | Traffic_3.mp4

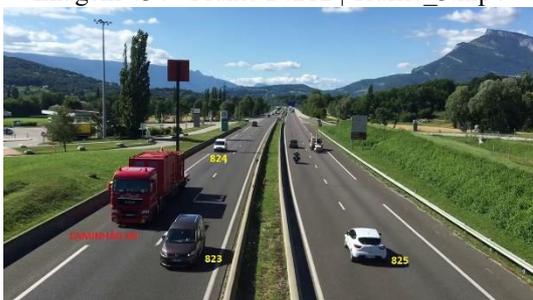


Imagem 435 - Frame 24238 | Traffic_3.mp4



Imagem 436 - Frame 24272 | Traffic_3.mp4



Imagem 437 - Frame 24334 | Traffic_3.mp4



Imagem 438 - Frame 24362 | Traffic_3.mp4



Imagem 439 - Frame 24394 | Traffic_3.mp4



Continua

Continuação

Imagem 440 - Frame 24430 | Traffic_3.mp4

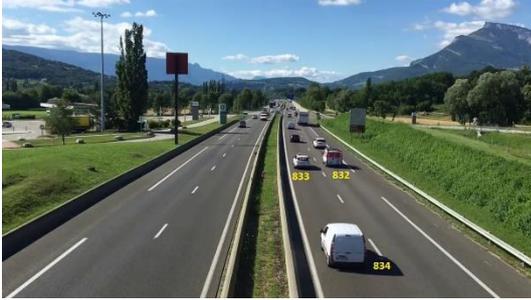


Imagem 441 - Frame 24454 | Traffic_3.mp4



Imagem 442 - Frame 24500 | Traffic_3.mp4

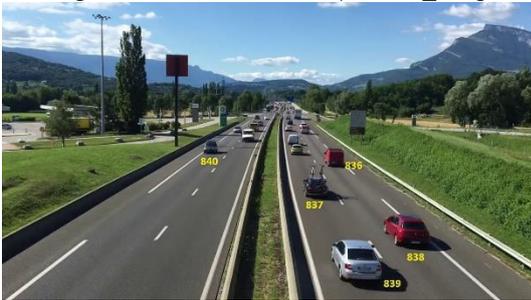


Imagem 443 - Frame 24544 | Traffic_3.mp4

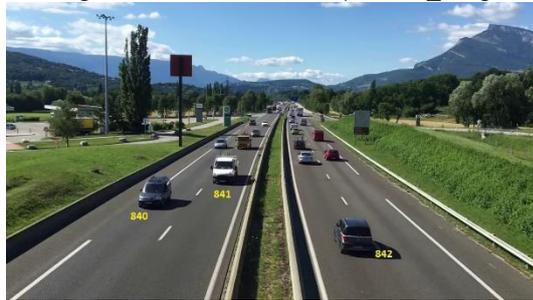


Imagem 444 - Frame 24600 | Traffic_3.mp4



Imagem 445 - Frame 24632 | Traffic_3.mp4

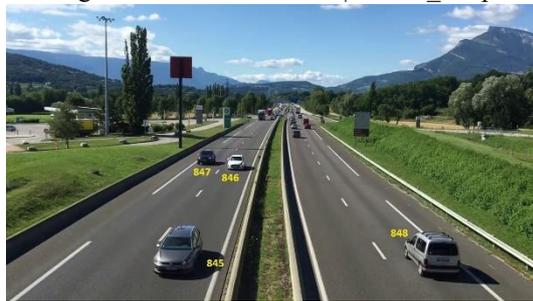
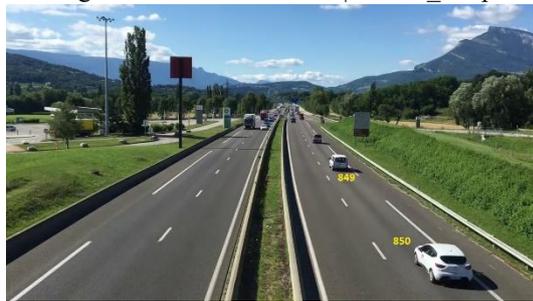


Imagem 446 - Frame 24676 | Traffic_3.mp4



Imagem 447 - Frame 24708 | Traffic_3.mp4



Continua

Continuação

Imagem 448 - Frame 24768 | Traffic_3.mp4



Imagem 449 - Frame 24824 | Traffic_3.mp4



Imagem 450 - Frame 24852 | Traffic_3.mp4



Imagem 451 - Frame 24878 | Traffic_3.mp4



Imagem 452 - Frame 24928 | Traffic_3.mp4



Imagem 453 - Frame 24970 | Traffic_3.mp4



Imagem 454 - Frame 25110 | Traffic_3.mp4



Imagem 455 - Frame 25138 | Traffic_3.mp4



Continua

Continuação

Imagem 456 - Frame 25178 | Traffic_3.mp4



Imagem 457 - Frame 25189 | Traffic_3.mp4



Fonte: O Autor

APÊNDICE F – Imagens de amostragem dos algoritmos quanto a precisão

Dados obtidos após aplicados os filtros a seguir, na base de dados disponível também no website informado anteriormente.

Quadro 22 - Imagens do controle - Teste 01 – GMG

Imagem 458 - Carro_Descendo_0141.png | GMG



Imagem 459 - Carro_Descendo_0171.png | GMG



Imagem 460 - Carro_Descendo_0259.png | GMG



Imagem 461 - Carro_Subindo_0295.png | GMG

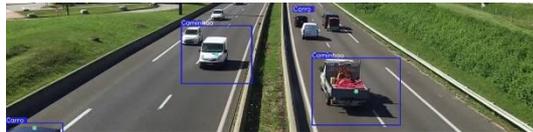


Imagem 462 - Carro_Descendo_0321.png | GMG



Imagem 463 - Caminhao_Subindo_0317.png | GMG



Imagem 464 - Caminhao_Subindo_0434.png | GMG

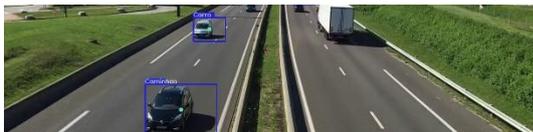


Imagem 465 - Caminhao_Subindo_0550.png | GMG

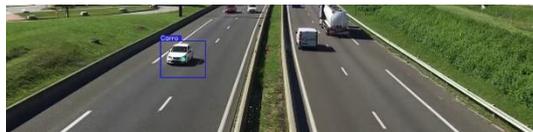


Imagem 466 - Caminhao_Descendo_0916.png | GMG



Imagem 467 - Caminhao_Descendo_1074.png | GMG



Fonte: O autor

Quadro 23 - Imagens do controle - Teste 01 – MOG

Imagem 468 - Carro_Subindo_0047.png | MOG



Imagem 469 - Carro_Descendo_0065.png | MOG



Imagem 470 - Carro_Descendo_0097.png | MOG



Imagem 471 - Carro_Descendo_0142.png | MOG



Imagem 472 - Carro_Descendo_0173.png | MOG



Imagem 473 - Caminhao_Subindo_0315.png | MOG



Imagem 474 - Caminhao_Subindo_0426.png | MOG



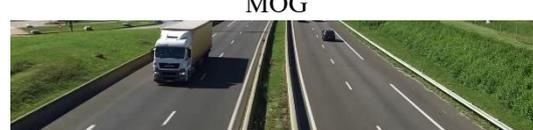
Imagem 475 - Caminhao_Subindo_0548.png | MOG



Imagem 476 - Caminhao_Subindo_1204.png | MOG



Imagem 477 - Caminhao_Descendo_1352.png | MOG



Fonte: O autor

Quadro 24 - Imagens do controle - Teste 01 – MOG2

Imagem 478 - Carro_Descendo_0095.png | MOG2



Imagem 479 - Carro_Descendo_0140.png | MOG2



Imagem 480 - Carro_Descendo_0172.png | MOG2



Imagem 481 - Carro_Descendo_0258.png | MOG2



Imagem 482 - Carro_Subindo_0296.png | MOG2



Imagem 483 - Caminhao_Descendo_0374.png | MOG2



Imagem 484 - Caminhao_Descendo_1349.png | MOG2



Imagem 485 - Caminhao_Descendo_1351.png | MOG2



Imagem 486 - Caminhao_Subindo_2113.png | MOG2



Imagem 487 - Caminhao_Descendo_2393.png | MOG2



Fonte: O autor

Quadro 25 - Imagens do controle - Teste 01 - KNN

Imagem 488 - Carro_Descendo_0049.png | KNN



Imagem 489 - Carro_Descendo_0088.png | KNN



Imagem 490 - Carro_Descendo_0132.png | KNN



Imagem 491 - Carro_Subindo_0126.png | KNN



Imagem 492 - Carro_Subindo_0246.png | KNN



Imagem 493 - Caminhao_Descendo_0251.png | KNN



Imagem 494 - Caminhao_Descendo_0287.png | KNN



Imagem 495 - Caminhao_Subindo_0342.png | KNN



Imagem 496 - Caminhao_Subindo_0448.png | KNN



Imagem 497 - Caminhao_Subindo_0450.png | KNN



Fonte: O autor

Quadro 26 - Imagens do controle - Teste 01 – CNT

Imagem 498 - Carro_Subindo_0045.png | CNT



Imagem 500 - Carro_Descendo_0096.png | CNT



Imagem 502 - Carro_Descendo_0141.png | CNT



Imagem 504 - Caminhao_Subindo_0332.png | CNT



Imagem 506 - Caminhao_Descendo_3267.png | CNT



Imagem 499 - Carro_Descendo_0065.png | CNT



Imagem 501 - Carro_Subindo_0109.png | CNT

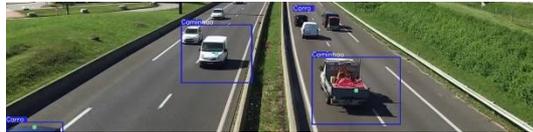


Imagem 503 - Caminhao_Subindo_0316.png | CNT

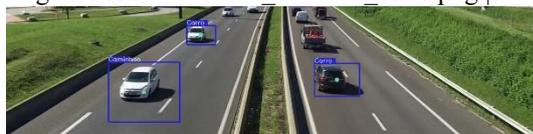


Imagem 505 - Caminhao_Descendo_1304.png | CNT

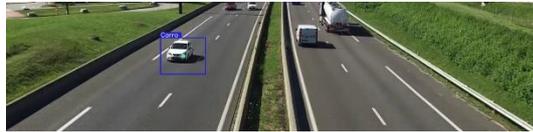


Imagem 507 - Caminhao_Subindo_4012.png | CNT



Fonte: O autor

Quadro 27 - Imagens do controle - Teste 01 – LSBP

Imagem 508 - Carro_Descendo_0086.png | LSBP

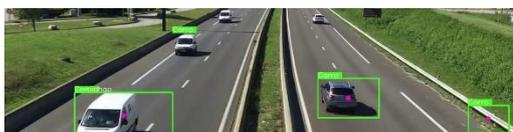


Imagem 510 - Carro_Descendo_0164.png | LSBP



Imagem 512 - Carro_Descendo_0250.png | LSBP



Imagem 514 - Caminhao_Subindo_0322.png | LSBP

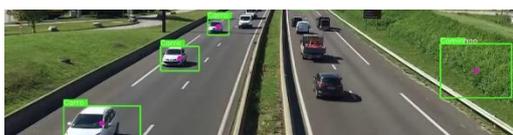


Imagem 516 - Caminhao_Subindo_0438.png | LSBP

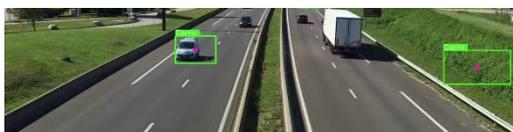


Imagem 509 - Carro_Descendo_0132.png | LSBP



Imagem 511 - Carro_Subindo_0172.png | LSBP

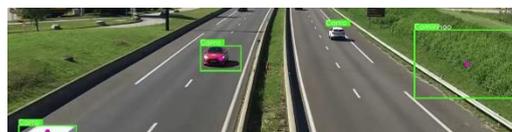


Imagem 513 - Caminhao_Descendo_0287.png | LSBP



Imagem 515 - Caminhao_Descendo_0369.png | LSBP



Imagem 517 - Caminhao_Subindo_0555.png | LSBP



Fonte: O autor

Quadro 28 - Imagens do controle - Teste 01 – GSOC

Imagem 518 - Carro_Descendo_0087.png | GSOC



Imagem 520 - Carro_Descendo_0132.png | GSOC



Imagem 522 - Carro_Descendo_0250.png | GSOC



Imagem 524 - Caminhao_Subindo_0555.png | GSOC



Imagem 526 - Caminhao_Descendo_1344.png | GSOC



Imagem 519 - Carro_Subindo_0118.png | GSOC



Imagem 521 - Carro_Descendo_0165.png | GSOC



Imagem 523 - Caminhao_Subindo_0438.png | GSOC

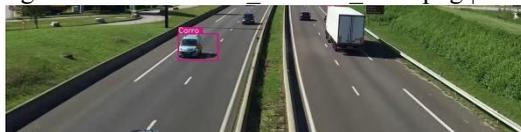


Imagem 525 - Caminhao_Subindo_1222.png | GSOC

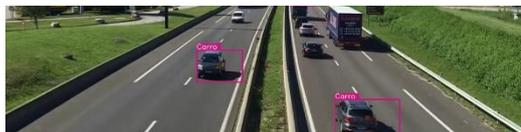


Imagem 527 - Caminhao_Subindo_2123.png | GSOC



Fonte: O autor

APÊNDICE G – Cessão de direitos

AUTOR: Rogério Santino Barboza

TÍTULO: Visão Computacional: estudo comparativo de algoritmos de subtração de fundo aplicados em soluções para o gerenciamento de tráfego urbano de veículos e pedestres.

GRAU: Mestre/2023

É concedida a Universidade Nove de Julho (UNINOVE) permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa dissertação de mestrado pode ser reproduzida sem autorização por escrito do autor ou citação da fonte.

Rogério Santino Barboza

São Paulo | SP | Brasil

rogerio.santino.barboza@gmail.com

ANEXO A – Dados sobre acidentes de trânsito - Brasil

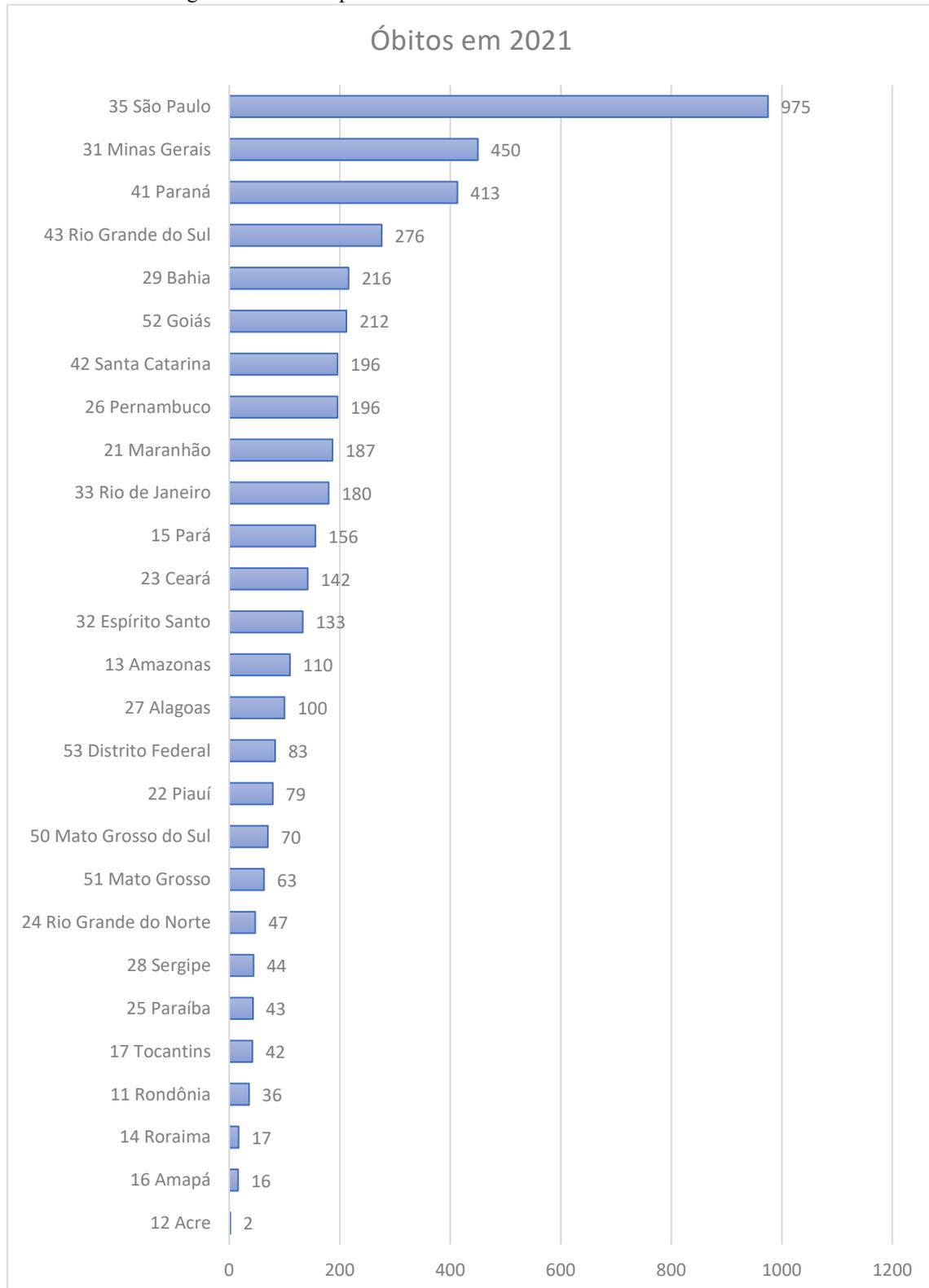
Dados obtidos após aplicados os filtros a seguir, na base de dados disponível em:
<http://tabnet.datasus.gov.br/cgi/deftohtm.exe?sim/cnv/pext10uf.def>

Filtros aplicados foram:

- **Linha:** Unidade da Federação
- **Coluna:** Ano do Óbito
- **Conteúdo:** Óbitos p/ Ocorrência
- **Períodos disponíveis:** 2021
- **Região:** Todas as categorias
- **Unidade da Federação:** Todas as categorias
- **Grande Grupo CID10:** V01-V99 Acidentes de transporte
- **Grupo CID10:** Pedestre traumatizado em um acidente de transporte
- **Categoria CID10:**
 - V02 Pedestre traum colis veic motor 2 3 rodas,
 - V03 Pedestre traum colis automov pickup caminhon,
 - V04 Pedestre traum colis veic transp pesado onib,
 - V09 Pedestre traum outr acid transp e NE
- **Faixa Etária:** Todas as categorias
- **Faixa Etária OPS:** Todas as categorias
- **Faixa Etária det:** Todas as categorias
- **Fx. Etária Menor 1A:** Todas as categorias
- **Sexo:** Todas as categorias
- **Cor/raça:** Todas as categorias
- **Escolaridade:** Todas as categorias
- **Estado civil:** Todas as categorias
- **Local ocorrência:** Todas as categorias
- **Acid. Trabalho:** Todas as categorias

O resultado obtido conforme demonstra a Figura 24 (p.276)

Figura 24 – Óbitos por causas externas conforme DATASUS – Brasil



Fonte: Dados extraídos do (DATASUS, 2022) e formatados pelo Autor

ANEXO B – Dados sobre a frota de veículos no Brasil

Tabela 21 - Frota de veículos no Brasil (2006-2021)

Indicador	2006	2007	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
Total	45.029.257	49.644.025	59.361.642	64.817.974	70.543.535	76.137.191	81.600.729	86.700.490	90.686.936	93.867.016	97.091.956	100.746.553	104.784.375	107.948.371	111.446.870
Automóvel	27.700.608	29.851.610	34.536.667	37.188.341	39.832.919	42.682.111	45.444.387	47.946.665	49.822.709	51.296.982	52.916.160	54.715.488	56.652.190	58.016.405	59.242.869
Bonde	216	210	137	126	93	62	48	46	45	44	43	42	42	42	42
Caminhão	1.761.659	1.847.225	2.026.269	2.143.467	2.274.947	2.380.780	2.488.680	2.588.984	2.645.992	2.684.227	2.720.548	2.766.097	2.826.343	2.879.080	2.947.856
Caminhão trator	278.468	304.918	367.189	412.473	457.954	492.640	541.118	578.765	593.892	606.679	623.112	655.047	697.732	735.748	796.166
Caminhonete	1.932.907	2.560.451	3.835.242	4.285.690	4.762.943	5.238.656	5.731.997	6.245.837	6.588.813	6.880.333	7.192.441	7.555.090	7.968.682	8.297.242	8.706.441
Camioneta	2.408.134	2.116.649	1.704.130	1.859.043	2.066.383	2.288.427	2.516.967	2.732.871	2.908.233	3.053.759	3.207.786	3.415.049	3.595.810	3.727.448	3.894.572
Chassi plataforma	7.079	6.913	4.620	3.801	3.261	2.972	2.291	2.079	1.942	1.877	1.806	1.765	1.722	1.700	1.672
Ciclomotor	83.306	83.615	89.350	95.259	107.919	127.324	142.857	159.705	223.755	338.470	370.361	387.458	404.343	415.291	430.347
Micro-Ônibus	196.578	214.483	249.016	270.693	296.761	318.762	340.928	361.501	375.274	383.325	390.235	398.839	412.046	420.105	426.229
Motocicleta	7.881.199	9.410.110	12.415.764	13.950.448	15.579.899	16.910.473	18.114.464	19.242.916	20.216.193	20.942.633	21.608.568	22.339.110	23.165.586	23.862.010	24.732.701
Motoneta	1.348.219	1.661.260	2.183.564	2.444.471	2.739.603	3.023.859	3.317.325	3.599.581	3.833.159	3.990.558	4.147.822	4.339.226	4.571.968	4.778.607	5.048.469
Ônibus	351.913	375.669	424.068	451.989	486.597	514.980	547.465	574.125	590.657	601.522	612.534	627.058	647.376	660.394	672.930
Quadriciclo	149	151	152	153	154	154	154	155	157	163	182	201	252	268	276
Reboque	525.876	573.344	679.968	782.421	866.508	961.795	1.070.606	1.185.242	1.296.184	1.399.298	1.501.219	1.608.078	1.722.890	1.852.654	1.993.536
Semi-reboque	455.998	497.454	588.327	612.040	670.309	722.572	789.614	843.404	873.106	896.354	920.504	960.352	1.016.350	1.075.545	1.154.738
Sidecar	6.600	7.825	8.405	8.418	8.491	8.505	8.509	8.510	8.519	8.524	8.524	8.525	8.526	8.542	8.561
Trator de esteira	78	94	115	155	179	203	208	208	208	208	208	208	211	211	211
Trator de rodas	13.227	14.656	18.711	21.679	24.666	27.026	28.363	29.516	30.372	30.896	31.363	31.948	32.973	33.894	34.934
Triciclo	2.305	2.881	6.417	10.258	14.838	19.052	22.615	25.518	28.417	30.903	32.769	34.795	36.934	38.595	40.503
Utilitário	71.139	111.524	216.415	269.217	340.747	407.685	482.027	563.861	637.211	707.152	791.763	887.042	1.005.535	1.125.451	1.290.380
Outros	3.599	2.983	7.116	7.832	8.364	9.153	10.106	11.001	12.098	13.109	14.008	15.135	16.864	19.139	23.437

Fonte: (IBGE, 2021)

ANEXO C – Dados associados sobre a cidade de São Paulo – SP - Brasil

Tabela 22 – Indicadores da cidade de São Paulo (a)

Fórmula:	Número total de acidentes com vítimas com automóvel / População total *100000										
Disponibilidade de dados:	Cidade, Subprefeitura, Distrito										
Fontes do Indicador:	a) SMS (Secretaria Municipal de Saúde) /SIVVA (Sistema de Informação e Vigilância de Violências e Acidentes) /COVISA (Coordenação de Vigilância em Saúde) b) IBGE (Instituto Brasileiro de Geografia e Estatística) - Projeção Populacional c) IBGE (Instituto Brasileiro de Geografia e Estatística)										
Observações:	Dados dos anos de atendimento 2008 e 2009 atualizados em 03/10/2014, de 2010 a 2014 atualizados em 03/10/2016 e de 2015 a 2017 atualizados em 18/03/2017										
Série Histórica de São Paulo											
Acidente com automóvel				Acidente com bicicleta				Acidente com motocicleta			
Número total de acidentes com vítimas com automóvel por cem mil habitantes.				Número total de acidentes com vítimas com bicicleta por cem mil habitantes.				Número total de acidentes com vítimas com motocicleta por cem mil habitantes.			
Período	Número total de acidentes com vítimas com automóvel	População total	Valor da Fórmula	Período	Número total de acidentes com vítimas com bicicleta	População total	Valor da Fórmula	Período	Número total de acidentes com vítimas com motocicleta	População total	Valor da Fórmula
2008	309	11.089.653	2,7864	2008	60	11.089.653	0,541	2008	470	11.089.653	4,2382
2009	1.832	11.171.578	16,3988	2009	376	11.171.578	3,3657	2009	3.055	11.171.578	27,3462
2010	1.932	11.253.503	17,168	2010	457	11.253.503	4,061	2010	3.414	11.253.503	30,3372
2011	1.768	11.303.626	15,641	2011	397	11.303.626	3,5121	2011	3.096	11.303.626	27,3894
2012	2.065	11.353.750	18,1878	2012	451	11.353.750	3,9723	2012	3.267	11.353.750	28,7746
2013	2.096	11.403.873	18,3797	2013	573	11.403.873	5,0246	2013	3.505	11.403.873	30,7352
2014	2.290	11.453.996	19,993	2014	473	11.453.996	4,1296	2014	3.724	11.453.996	32,5127
2015	1.990	11.504.120	17,2982	2015	416	11.504.120	3,6161	2015	2.890	11.504.120	25,1214
2016	1.829	11.554.243	15,8297	2016	442	11.554.243	3,8254	2016	3.145	11.554.243	27,2194
2017	2.835	11.696.088	24,2389	2017	862	11.696.088	7,37	2017	6.034	11.696.088	51,5899
FONTE: (REDE SOCIAL BRASILEIRA POR CIDADES JUSTAS E SUSTENTÁVEIS (A), 2022)				FONTE: (REDE SOCIAL BRASILEIRA POR CIDADES JUSTAS E SUSTENTÁVEIS (B), 2022)				FONTE: (REDE SOCIAL BRASILEIRA POR CIDADES JUSTAS E SUSTENTÁVEIS (C), 2022)			

Tabela 23 – Indicadores da cidade de São Paulo (b)

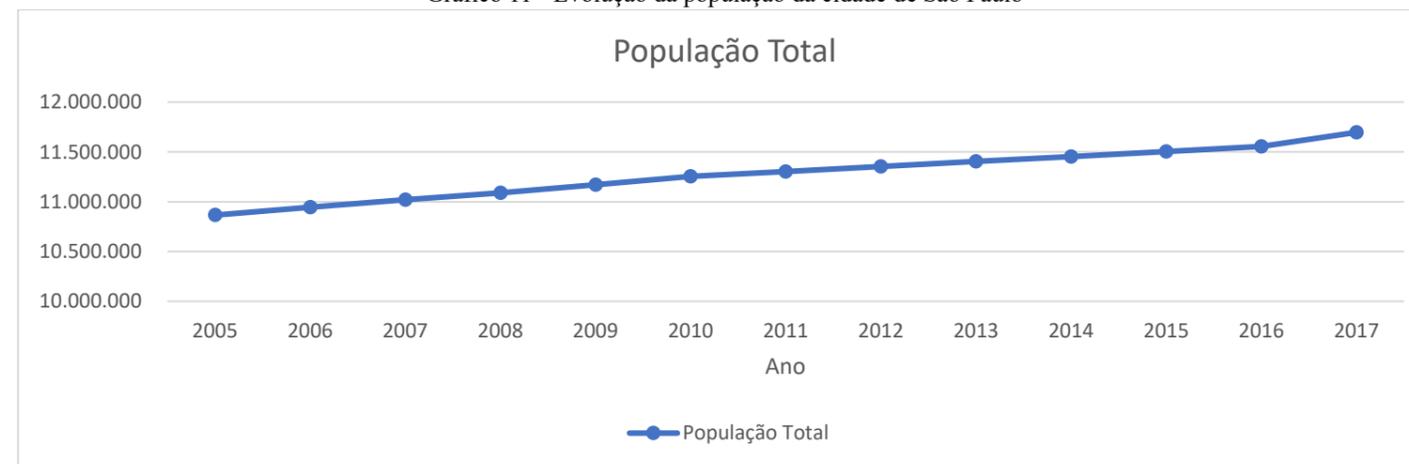
Fórmula:	Número total de acidentes com vítimas com automóvel / População total *100000										
Disponibilidade de dados:	Cidade, Subprefeitura, Distrito										
Fontes do Indicador:	a) SMS (Secretaria Municipal de Saúde) /SIVVA (Sistema de Informação e Vigilância de Violências e Acidentes) /COVISA (Coordenação de Vigilância em Saúde) b) IBGE (Instituto Brasileiro de Geografia e Estatística) - Projeção Populacional c) IBGE (Instituto Brasileiro de Geografia e Estatística) d) Planilhão 2017- SAMU-SP										
Observações:	Dados dos anos de atendimento 2008 e 2009 atualizados em 03/10/2014, de 2010 a 2014 atualizados em 03/10/2016 e de 2015 a 2017 atualizados em 18/03/2017										
Série Histórica de São Paulo											
Acidentes de trânsito				Atropelamentos				Mortes por atropelamento			
Número total de acidentes de trânsito por 10 mil habitantes.				Número total de atropelamentos por 10 mil habitantes.				Número de mortes por atropelamentos, por 10 mil habitantes			
Período	Número total de acidentes de trânsito	População total	Valor da Fórmula	Período	Número total de atropelamentos	População total	Valor da Fórmula	Período	Total de mortes por atropelamentos	População total	Valor da Fórmula
2008	2.690	11.089.653	2,4257	2008	683	11.089.653	0,6159	2005	10.865.573	701	0,6452
2009	8.816	11.171.578	7,8915	2009	2.048	11.171.578	1,8332	2006	10.944.889	689	0,6295
2010	10.347	11.253.503	9,1945	2010	2.473	11.253.503	2,1975	2007	11.019.484	632	0,5735
2011	9.472	11.303.626	8,3796	2011	2.236	11.303.626	1,9781	2008	11.089.653	622	0,5609
2012	11.560	11.353.750	10,1817	2012	2.833	11.353.750	2,4952	2009	11.171.578	578	0,5174
2013	12.212	11.403.873	10,7086	2013	3.002	11.403.873	2,6324	2010	11.253.503	503	0,447
2014	8.577	11.453.996	7,4882	2014	3.684	11.453.996	3,2163	2011	11.303.626	514	0,4547
2015	7.343	11.504.120	6,3829	2015	2.780	11.504.120	2,4165	2012	11.353.750	573	0,5047
2016	7.404	11.554.243	6,408	2016	2.693	11.554.243	2,3307	2013	11.403.873	509	0,4463
2017	10.107	11.696.088	8,6414	2017	2.631	11.696.088	2,2495	2014	11.453.996	565	0,4933
FONTE: (REDESOCIALDECIDADES.ORG.BR (D), 2022)				FONTE: (REDE SOCIAL BRASILEIRA POR CIDADES JUSTAS E SUSTENTÁVEIS (F), 2022)				FONTE: (REDE SOCIAL BRASILEIRA POR CIDADES JUSTAS E SUSTENTÁVEIS (D), 2022)			

Tabela 24 - Dados associados da Cidade de São Paulo

Período	População Total	Total de mortes por atropelamentos	Número de mortes em acidentes de trânsito	Total de mortes de ocupantes de motocicleta	Total de mortes com bicicleta	Total de mortes de ocupantes de automóveis e caminhonetes	Número total de atropelamentos	Número total de acidentes de trânsito	Número total de acidentes com vítimas com motocicleta	Número total de acidentes com vítimas com bicicleta	Número total de acidentes com vítimas com automóvel
2005	10.865.573	701		282	75	221					
2006	10.944.889	689	1.331	314	83	194					
2007	11.019.484	632	1.426	255	42	118					
2008	11.089.653	622	1.283	376	52	166	683	2.690	470	60	309
2009	11.171.578	578	1.237	303	45	144	2.048	8.816	3.055	376	1.832
2010	11.253.503	503	1.143	395	44	139	2.473	10.347	3.414	457	1.932
2011	11.303.626	514	1.164	433	46	134	2.236	9.472	3.096	397	1.768
2012	11.353.750	573	1.458	416		256	2.833	11.560	3.267	451	2.065
2013	11.403.873	509	1.334	349		284	3.002	12.212	3.505	573	2.096
2014	11.453.996	565	1.410	392	46	253	3.684	8.577	3.724	473	2.290
2015	11.504.120	432	1.139	360	35	231	2.780	7.343	2.890	416	1.990
2016	11.554.243		854	317	30	204	2.693	7.404	3.145	442	1.829
2017	11.696.088		796				2.631	10.107	6.034	862	2.835

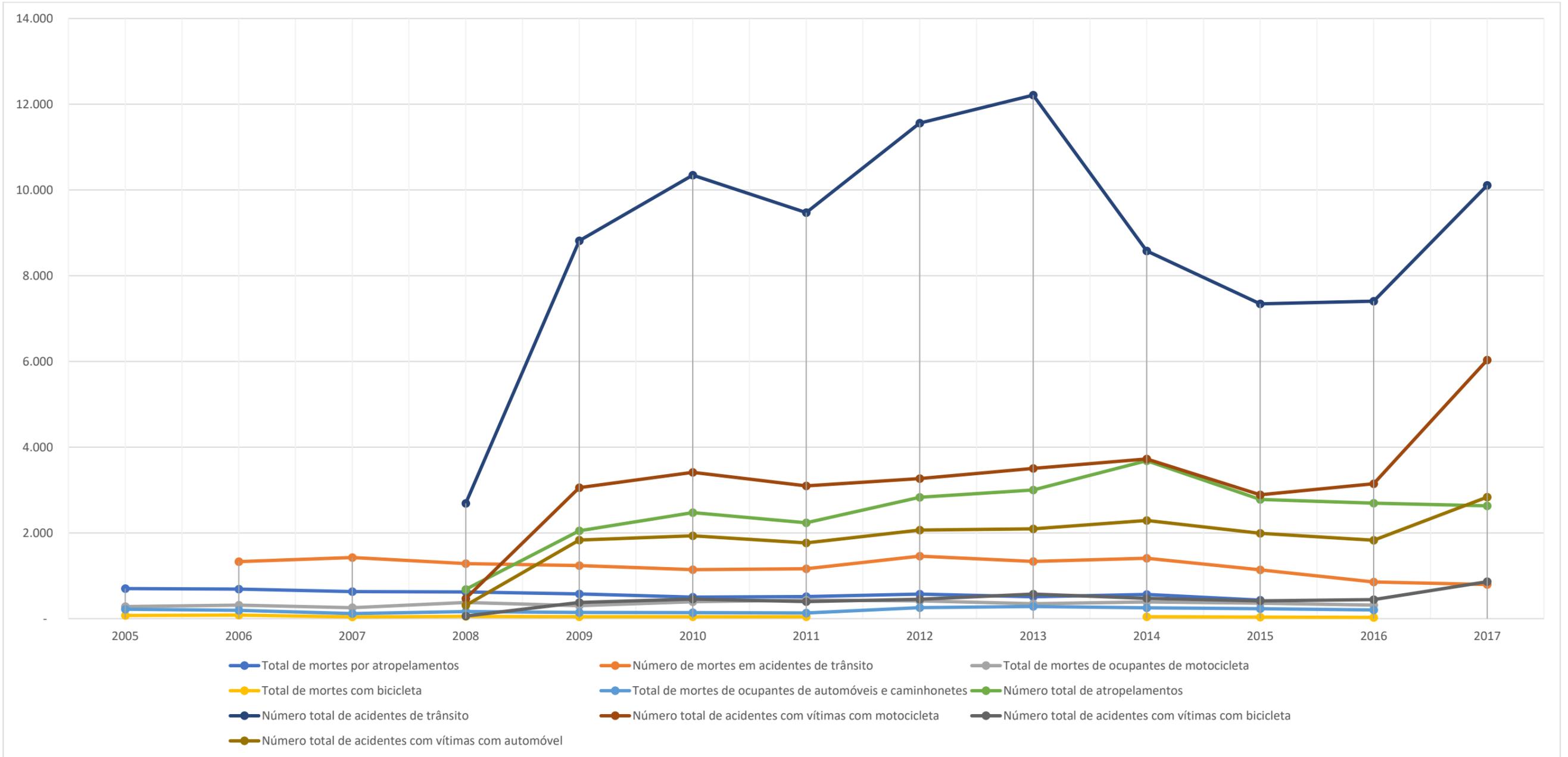
Fonte: o autor com base nos dados fornecidos por Rede Social de Cidades - 2022

Gráfico 11 - Evolução da população da cidade de São Paulo



Fonte: o autor com base nos dados fornecidos por Rede Social de Cidades - 2022

Gráfico 12 - Indicadores da cidade de São Paulo



Fonte: O autor, com base nos dados fornecidos por Rede Social de Cidades - 2022