

**UNIVERSIDADE DO EXTREMO SUL CATARINENSE - UNESC  
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**RAFAEL MICHELS AMÉRICO**

**CONTROLE DE VAGAS DE ESTACIONAMENTO A PARTIR DE TRATAMENTO  
DE IMAGENS EM TEMPO REAL COM A BIBLIOTECA OPENCV**

**CRICIÚMA  
2019**

**RAFAEL MICHELS AMÉRICO**

**CONTROLE DE VAGAS DE ESTACIONAMENTO A PARTIR DE TRATAMENTO  
DE IMAGENS EM TEMPO REAL COM A BIBLIOTECA OPENCV**

Trabalho de Conclusão de Curso, apresentado para obtenção do grau de Bacharel no curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC.

Orientador: Prof. Esp. Sérgio Coral

**CRICIÚMA**

**2019**


**RAFAEL MICHELS AMÉRICO**

**CONTROLE DE VAGAS DE ESTACIONAMENTO A PARTIR DE TRATAMENTO  
DE IMAGENS EM TEMPO REAL COM A BIBLIOTECA OPENCV**


Trabalho de Conclusão de Curso aprovado pela Banca Examinadora para obtenção do Grau de Bacharel, no Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC, com Linha de Pesquisa em Automação de Sistema.

Criciúma, 06 de Dezembro de 2019.

**BANCA EXAMINADORA**

  
Prof. Esp. Sérgio Coral - UNESC - Orientador

  
Prof. Esp. Gilberto Vieira da Silva - UNESC

  
Prof. Esp. Giacomino Antônio Althoff Bolan - UNESC

## **AGRADECIMENTOS**

Primeiramente agradeço a Deus, por proporcionar inúmeras bênçãos e fornecer a sabedoria necessária para percorrer os caminhos corretos, a fim de alcançar meus objetivos e concluir mais esta etapa da minha vida.

Aos meus pais Eraldo Américo e Adriana Bortoluzzi Michels Américo, que sempre prestaram um apoio incondicional em todos os momentos, ao qual mostro-me o quanto é importante batalhar por algo que se deseja. A eles todo o meu amor e agradecimento.

A minha namorada Laís Martinello, que foi uma das principais pessoas a acreditar no meu potencial durante este período e que sempre prestou todo tipo de apoio em tudo que fosse necessário, principalmente me dando forças e fazendo com que eu persistisse na luta pelos meus objetivos. A ela todo meu amor e carinho.

Aos meus colegas de trabalho André Rossi e Gustavo, que sempre me incentivaram, acreditaram em minha capacidade para o desenvolvimento deste projeto e me acompanharam durante esta etapa.

Um agradecimento em especial ao meu orientador Sérgio Coral, que não mediu esforços para me auxiliar em tudo que fosse necessário para que eu conseguisse obter sucesso nesta etapa. A ele meu eterno agradecimento.

A UNESC e a todos os professores do curso de Ciência da Computação, que compartilharam seus conhecimentos durante esta jornada e por proporcionarem um ótimo ambiente para o aprendizado, bem como profissionais qualificados e dedicados na missão de repassar os seus conhecimentos a todos os acadêmicos. A todos vocês meu muito obrigado.

**“O maior sucesso é feito da liberdade de fracassar.”**

**Mark Zuckerberg**

## RESUMO

O crescimento desordenado de veículos nas ruas é um grande problema em todos os países, e vem aumentando significativamente com o passar dos anos, gerando grandes congestionamentos em vias e afetando também na procura por locais para estacionar o veículo. Com a evolução da tecnologia surgiram soluções para amenizar o transtorno pela procura de uma vaga de estacionamento, porém não são muitas opções, e possuem um alto custo de implantação. Diante deste contexto, o presente trabalho visa a pesquisa e o desenvolvimento de um protótipo capaz de identificar as vagas de estacionamento utilizando conceitos de visão computacional, aplicando o tratamento de imagens com auxílio da biblioteca OpenCV; biblioteca está específica para este tipo de ramo de aplicação em processamento de imagens com licença de código aberto. O software desenvolvido na linguagem Python captura a imagem do estacionamento em tempo real através de uma câmera IP e este aplica um processamento de imagens com auxílio de filtros e métodos que compõem a biblioteca, sendo possível definir as vagas de estacionamento através de pontos da imagem e estes interpretados pelo código fonte para então apresentá-los. A identificação dos carros para alteração do *status* das vagas foi obtido através do método de *Laplace* que também faz parte da biblioteca. A utilização frequente de *smartphone* no dia-a-dia tornou-se uma ferramenta indispensável devido aos recursos que ele oferece, também é proposto desenvolvimento de uma página WEB para que o usuário possa visualizar as vagas disponíveis e ocupadas.

**Palavras-chave:** Visão Computacional, Processamento de imagens, OpenCV, Estacionamento, WEB.

## ABSTRACT

The disordered vehicle growth on the streets is a huge problem all around the world, and it keeps increasing significantly over the years, causing traffic jams and making it hard to find a parking spot. With technology evolution solutions to ease the inconvenience when looking for a place to park appeared, but there aren't many options and there's a high implementation cost. Given the context, this paper aims the research and the development of a prototype capable of identifying parking spots deploying computer vision concepts, applying image handling with the aid of OpenCV library; a specific library to this kind of field which applies in image processing with open code license. The software developed in Python language captures the parking lot image in real time through an IP camera and it applies an image processing with the assistance of filters and other methods that are part of the library, being possible to define the parking spaces through spots in the image and these are understood by the source code and then it introduces them. Cars identification to change the spot status was obtained through Lapalace method which is also part of the library. The frequent smartphone use in our daily lives became an essential tool due to the resources provided. In addition, it is proposed a WEB page development so the users are able to see available and taken spots.

**Keywords:** Computer Vision, Image Processing, OpenCV, Parking, WEB.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Demanda global de vendas de veículos leves 1995-2018 .....	10
Figura 2 – Excesso de veículos na Zona Sul de São Paulo .....	12
Figura 3 – Estacionamento vertical automatizado .....	13
Figura 4 – Estacionamento inteligente shopping Morumbi .....	14
Figura 5 – Fluxo dos elementos de um sistema PID .....	16
Figura 6 – Etapas de um sistema de visão computacional .....	17
Figura 7 – Modelo RGB .....	20
Figura 8 – Efeito do filtro da média em uma imagem .....	21
Figura 9 – Representação de uma imagem após utilizado o método : <i>cvFindContours()</i> .....	23
Figura 10 – Representação matemática do método de Laplace .....	24
Figura 11 – Estrutura de um arquivo YAML e XML .....	25
Figura 12 – Diferença entre câmeras IP e câmeras analógicas .....	28
Figura 13 – Representação de uma comunicação com câmeras IP .....	29
Figura 14 – Exemplo de uma infraestrutura de dispositivos móveis .....	30
Figura 15 – Representação de uma câmera IP .....	38
Figura 16 – Representação das vagas com os pontos definidos .....	42
Figura 17 – Detecção de bordas com o método Laplace .....	43
Figura 18 – Implementação do método Laplace .....	43
Figura 19 – Configuração de acesso ao banco de dados na linguagem Python .....	45
Figura 20 – Configuração de acesso ao banco de dados na linguagem PHP .....	46
Figura 21 – Modelagem do banco de dados .....	46
Figura 22 – Comando SQL na linguagem Python .....	47
Figura 23 – Comando SQL na linguagem PHP .....	47
Figura 23 – Comparativo da taxa de acerto da identificação de pontos que correspondem à áreas de estacionamento .....	50
Figura 23 – Comparativo da taxa de acerto da identificação da área que correspondem à áreas de estacionamento .....	50

## LISTA DE TABELAS

Tabela 1 - Equipamentos utilizados para a montagem da simulação .....	39
Tabela 2 – Tabela comparativa das taxas de erros e acertos .....	49
Tabela 3 – Resultados dos testes realizados .....	49

## LISTA DE ABREVIATURAS E SIGLAS

CENTAC	Central de Atendimento ao Acadêmico
CCD	<i>Charge Coupled Device</i>
CMYK	Padrão de cores formado por Ciano ( <i>Cyan</i> ), Magenta ( <i>Magenta</i> ), Amarelo ( <i>Yellow</i> ) e Preto ( <i>Black (Key)</i> ).
DETRAN	Departamento Estadual de Trânsito do Estado de Santa Catarina
GPS	<i>Global Positioning System</i>
HD	<i>High Definition</i>
HTTP	<i>Hyper Text Transfer Protocol</i>
IP	<i>Internet Protocol</i>
LED	Diodo emissor de luz ( <i>Light Emitting Diode</i> ).
OpenCV	<i>Open-source Computer Vision Library</i>
RGB	Padrão de cores formado por Vermelho ( <i>Red</i> ), Verde ( <i>Green</i> ) e Azul ( <i>Blue</i> ).
UNESC	Universidade do Extremo Sul Catarinense

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>6</b>
1.1 JUSTIFICATIVA .....	7
1.2 OBJETIVO GERAL .....	8
1.3 OBJETIVOS ESPECÍFICOS .....	8
1.4 ESTRUTURA DO TRABALHO .....	9
<b>2 FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>10</b>
2.1 MOBILIDADE URBANA .....	10
<b>2.1.1 Setor automobilístico</b> .....	<b>10</b>
<b>2.1.2 Problemas na atualidade</b> .....	<b>11</b>
<b>2.1.3 Estacionamento</b> .....	<b>12</b>
<b>2.1.4 Estacionamento UNESCO</b> .....	<b>14</b>
2.2 PROCESSAMENTO DE IMAGENS .....	15
<b>2.2.1 Visão Computacional</b> .....	<b>16</b>
<b>2.2.2 Imagens digitais</b> .....	<b>18</b>
<b>2.2.3 Formatos de arquivos</b> .....	<b>18</b>
<b>2.2.4 Espaço cor</b> .....	<b>19</b>
<b>2.2.5 Aplicação de filtros</b> .....	<b>20</b>
2.3 OPENCV .....	21
<b>2.3.1 Contornos</b> .....	<b>22</b>
<b>2.3.2 Laplace</b> .....	<b>23</b>
<b>2.3.3 YAML</b> .....	<b>24</b>
2.4 COMUNICAÇÃO DE DADOS .....	25
<b>2.4.1 Protocolos de comunicação</b> .....	<b>26</b>
2.4.1.1 TCP .....	26
2.5 CÂMERAS IP .....	27
<b>2.5.1 Comunicação câmera IP</b> .....	<b>28</b>
<b>2.5.2 Infravermelho</b> .....	<b>29</b>
<b>2.5.3 Resolução</b> .....	<b>29</b>
2.6 DISPOSITIVOS MÓVEIS .....	30
2.7 NAVEGADORES .....	31
<b>2.7.1 WEBAPPS</b> .....	<b>31</b>
<b>3 TRABALHOS CORRELATOS</b> .....	<b>33</b>

3.1 EM RUMO A UM SISTEMA AUTOMÁTICO DE CONTROLE DE ACESSO DE VEÍCULOS AUTOMOTIVOS: RECONHECIMENTO DE CARACTERES EM PLACAS DE VEÍCULOS .....	33
3.2 ESTACIONAMENTO INTELIGENTE .....	33
3.3 RECONHECIMENTO AUTOMÁTICO DE PLACAS VEICULARES UTILIZANDO A TECNOLOGIA OCR E A PLATAFORMA RASPBERRY PI APLICADA NA FISCALIZAÇÃO ELETRÔNICA DE RODOVIAS.....	34
3.4 USO DE VISÃO COMPUTACIONAL EM DISPOSITIVOS MÓVEIS PARA AUXÍLIO À TRAVESSIA DE PEDESTRES COM DEFICIÊNCIA VISUAL .....	35
3.5 PLANEJAMENTO DE TRAJETÓRIA PARA ESTACIONAMENTO DE VEÍCULOS AUTÔNOMOS.....	35
3.6 PROCESSO DE RECONHECIMENTO NÃO SUPERVISIONADO DE ÁREAS DE ESTACIONAMENTO.....	36
<b>4 CONTROLE DE VAGAS DE ESTACIONAMENTO A PARTIR DE TRATAMENTO DE IMAGENS EM TEMPO REAL COM A BIBLIOTECA OPENCV .....</b>	<b>37</b>
4.1 METODOLOGIA.....	37
4.1.1 Ferramentas e levantamento de requisitos.....	38
4.1.2 Fluxo de dados e modelo de comunicação .....	39
4.1.3 Desenvolvimento do protótipo.....	40
4.1.4 Desenvolvimento da página WEB.....	44
4.1.5 Integração entre software e aplicação web.....	45
4.1.6 Funcionamento da aplicação web .....	47
4.2 RESULTADOS E DISCUSSÕES .....	48
<b>5 CONCLUSÃO .....</b>	<b>52</b>
<b>REFERÊNCIAS.....</b>	<b>54</b>

## 1 INTRODUÇÃO

O avanço da tecnologia proporcionou velocidade e dinamismo em tarefas realizadas no dia-a-dia para as pessoas. Por exemplo, ao pegar um ônibus ou outro meio de locomoção para um determinado destino pode haver o envolvimento da tecnologia, assim, pode se dizer que a influência desta no ritmo de vida da sociedade atual é enorme (MONTEIRO; VACCARI, 2016).

Com o aumento da necessidade da locomoção mais ágil e eficiente das pessoas para locais de trabalho, moradia, lazer, entre outros locais que fazem parte da rotina diária, proporcionou um grande aumento dos veículos individuais e particulares em circulação nas cidades. Em algumas situações de locomoção se é necessário que o condutor pegue as vias mais cedo, horas antes do planejado. Esta necessidade se dá por conta dos grandes engarrafamentos e a dificuldade de encontrar uma vaga de estacionamento (NETO, 2018).

A procura por vaga de estacionamento nos grandes centros é diária, muitos condutores optam por utilizar um serviço de estacionamento privado, onde deixam seu carro enquanto realizam suas tarefas diárias, porém, ainda assim encontram estacionamentos lotados, devido ao grande fluxo de carros nos estacionamentos (HAMILTON, 2017).

Segundo Medeiros (2018) 92% dos brasileiros possuem ou usam um smartphone. Os smartphones se tornaram uma ferramenta essencial para as pessoas, além de possuírem grande poder de processamento integram vários recursos. Estes dispositivos possibilitam que as pessoas realizem ligações, conexão com a internet com alta disponibilidade e a utilização de aplicativos que proporcionam agilidade em suas tarefas diárias, tais como GPS, aplicativos bancários, entre outras funções que o dispositivo pode proporcionar (RIBEIRO, 2018).

O desenvolvimento de aplicativos na área de visão computacional conta na maioria das vezes com algum tipo de biblioteca, tendo objetivo de auxiliar como um componente chave baseado na visão de aplicações. Existem inúmeras bibliotecas, cada uma delas com suas características, muitas delas são complicadas ou limitadas à compatibilidade de hardware/software, estes fatores tendem a impedir o desenvolvimento de aplicações de pesquisa (GRANGE; FONG; BAUR, 2003, tradução nossa).

A biblioteca OpenCV, aplicada para o desenvolvimento do protótipo no presente trabalho, é um componente multiplataforma, totalmente livre sem fins lucrativos, podendo ser aplicada para fins acadêmicos quanto comercialmente ou para qualquer pessoa tenha o interesse em utilizá-la (BRADSKI; KAEHLER, 2008, tradução nossa).

A Visão Computacional é uma área da computação que estuda métodos para a extração de informações dos objetos em uma imagem, como velocidade, formas, entre outras. Todas as extrações são feitas para alcançar um determinado objetivo. Diferente do processamento de imagens, esta área consiste em transformar uma imagem em outra imagem, através da manipulação e obtenção dos dados de uma imagem e do uso destas para diferentes objetivos (RIOS, 2010).

## 1.1 JUSTIFICATIVA

Com o crescimento desordenado de veículos em circulação nas áreas urbanas surgem inúmeros problemas, como conflitos entre condutores de veículos automotores e a falta de espaços destinados para estacionar. Segundo DETRAN-SC através de dados estatísticos a frota de veículos em 2015 na cidade de Criciúma era de 1.697.092 veículos, e em 2018 o número de veículos saltou para 1.801.086 (SANTA CATARINA, 2018).

O presente trabalho será desenvolvido com o intuito de amenizar o transtorno de encontrar uma vaga disponível em estacionamentos, em especial, da UNESC, onde alguns dos estacionamentos nem mesmo tem as vagas demarcadas, resultando em uma má utilização dos estacionamentos.

Inicialmente surgiu a ideia de utilização de sensores em cada vaga de estacionamento, para o reconhecimento de vagas disponíveis e ocupadas. Através de pesquisas feitas sobre este tipo de monitoramento, constatou-se que esta solução é inviável pelo alto valor de investimento em sensores de monitoramento para cada vaga e a infraestrutura de segurança para funcionamento seguro da solução (SELVENCA; COUTO, 2013).

A utilização de câmeras IP para o monitoramento de segurança cresceu nos últimos anos, pois estas possibilitaram uma nova visão no setor de segurança, onde este tipo de equipamento pode ser acessado de qualquer lugar desde que o usuário possua uma conexão com a internet e permissão de acesso. A vantagem de

utilizar uma câmera IP para monitoramento de vaga de estacionamento, é que com apenas um equipamento pode-se monitorar diversas vagas sem necessitar de instalação direta de uma câmera sobre cada vaga (SELVENCA; COUTO, 2013).

Optou-se pelo desenvolvimento de uma página WEB para a visualização das vagas disponíveis e ocupadas, devido a sua alta compatibilidade com computadores, notebooks e dispositivos móveis. (NESBITT, 2018, tradução nossa).

A utilização da biblioteca OpenCV é de grande importância quando se trata no desenvolvimento de sistemas na área de visão computacional. Uma das vantagens é sua arquitetura e gerenciamento de memória. Este componente fornece uma estrutura para trabalhar com imagens e vídeos em tempo real, além disso, disponibiliza uma grande quantidade de algoritmos prontos (BRAHMBHATT, 2013, tradução nossa).

O desenvolvimento do presente projeto será analisado a melhor maneira de tratar as imagens capturadas pela câmera IP a fim de detectar as de um determinado estacionamento e de uma página WEB será desenvolvida para a visualização das vagas.

## 1.2 OBJETIVO GERAL

Desenvolver um protótipo utilizando a biblioteca do OpenCV a fim de identificar vagas disponíveis no estacionamento da UNESC através de imagens capturadas por uma câmera de monitoramento, e uma página WEB para visualização da localização das vagas disponíveis.

## 1.3 OBJETIVOS ESPECÍFICOS

Os objetivos específicos deste trabalho estão listados a seguir:

- a) descrever o conceito de visão computacional e funcionamento de tratamento de imagens em tempo real;
- b) verificar a melhor maneira para identificação de vagas do estacionamento;
- c) desenvolver um protótipo com auxílio da biblioteca OpenCV para reconhecimento das vagas de estacionamento

d) desenvolver uma página WEB para visualização das vagas de estacionamento;

;

#### 1.4 ESTRUTURA DO TRABALHO

Este trabalho de pesquisa é dividido em quatro capítulos, sendo que o primeiro capítulo aborda brevemente o que será exibido no trabalho de pesquisa, e sua estrutura é formada pela introdução, objetivo geral e seus objetivos específicos, e justificativa para a qual o projeto será desenvolvido.

O segundo capítulo é composto pelo levantamento bibliográfico, onde o autor aborda o conceito de mobilidade urbana, o crescimento do setor automobilístico, os problemas existentes na atualidade e também sobre estacionamentos.

No terceiro capítulo foi abordado os estudos a processamento de imagens, onde foi explanado sobre visão computacional, conceito, imagens digitais, espaço cor, formatos de arquivos, ilustrando o funcionamento para processar uma imagem a fim de extrair informações dela, juntamente com explicações sobre a biblioteca OpenCV, que será utilizada no desenvolvimento do protótipo para reconhecimento de vagas de estacionamento, sobre a comunicação de dados e o protocolo de comunicação TCP. As câmeras IP, no qual será utilizada para monitoramento do estacionamento e também sobre os navegadores WEB.

E por fim, o terceiro capítulo aborda os trabalhos correlatos a este trabalho de pesquisa em questão.

Finalizando com o quarto capítulo, onde é explicado a respeito do trabalho proposto e a metodologia que será utilizada para tal qual. Este capítulo é concluído demonstrando os recursos necessários para o desenvolvimento do trabalho em questão.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 MOBILIDADE URBANA

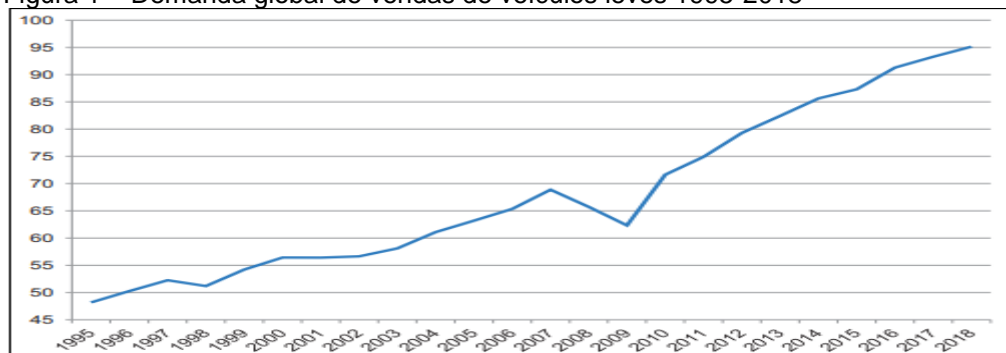
A mobilidade urbana é um tema importante quando se discute sobre o desenvolvimento urbano e qualidade de vida da população. Há diversos conceitos que a definem, dentre eles podemos citar o deslocamento das pessoas e das mercadorias nos centros urbanos, sendo este realizado por meio de transportes públicos ou privados (RODRIGUES, 2017, tradução nossa).

No século XX, o automóvel se tornou uma figura importante na cultura, da sociedade e da economia mundial. Mas com o passar do tempo o número de automóveis teve um crescimento de forma significativa, e, sem um aumento proporcional no número das estradas, surgiram os grandes engarrafamentos no século seguinte, e assim aumentando também a poluição nas cidades (BORGES, 2012).

#### 2.1.1 Setor automobilístico

O mercado global de veículos leves (carros e motos) obteve um recorde de vendas em 2017, totalizando oito anos consecutivos de crescimento. Especialistas previram que em 2018 o mercado de uma desaceleração de 2% nas vendas, devido às altas taxas, totalizando em aproximadamente 95 milhões de unidades vendidas em todo o mundo, o gráfico na figura 1, ilustra o grande crescimento das vendas mundiais de automóveis durante os anos (STOREY, 2017, tradução nossa).

Figura 1 – Demanda global de vendas de veículos leves 1995-2018



Fonte: Automotive World (2017).

No Brasil, o mercado automobilístico obteve um crescimento de 26,6% de veículos vendidos em fevereiro de 2019, total de 198.653 automóveis. Comparação feita com fevereiro de 2018, onde foram vendidos 156.880 veículos. Segundo a associação de concessionárias, a Fenabrave, o primeiro bimestre do ano de 2019, 398.441 veículos foram vendidos no Brasil, obtendo um crescimento de 17,8%, comparado ao mês de janeiro e fevereiro de 2018. No ano anterior, as concessionárias venderam 338.125 unidades (G1, 2019).

No estado de Santa de Catarina segundo DETRAN-SC, através de dados estatísticos a frota de veículos no ano de 2015 foi de 53.915.892 veículos, e no ano de 2018 um total de 59.258.919 (SANTA CATARINA, 2018).

### **2.1.2 Problemas na atualidade**

Ao longo dos anos o espaço urbano vem passando por diversas transformações. Com os locais de trabalho e lazer cada vez mais distantes das pessoas e a busca da facilidade e comodidade na locomoção individual, a quantidade de automóveis nas vias vem crescendo gradualmente. Com este crescimento surgiram fatores prejudiciais. As vias, por exemplo, não conseguem acompanhar o crescimento desenfreado da frota de automóveis, criando grandes congestionamentos nas cidades com maior volume de automóveis (CARVALHO, 2016).

Segundo Borges (2012) a cultura de obter um automóvel particular e a ineficiência dos transportes públicos em atender os usuários são os maiores responsáveis pelos conflitos de mobilidade urbana da atualidade. A falta de transporte público com preço justo e de qualidade faz com que a população adere pelo transporte individual, e assim agravando aos grandes problemas relacionados ao trânsito.

A figura 2 ilustra as consequências do excesso de automóveis nas ruas.

Figura 2 – Excesso de veículos na Zona Sul de São Paulo



Fonte: El País (2017).

### 2.1.3 Estacionamento

Um grande fator que gera desconforto para a população além dos citados anteriormente, é a procura de um local adequado para estacionar o veículo.

Estacionamentos são áreas destinadas para que os motoristas possam estacionar seus carros de maneira segura e tranquila, para facilitar o acesso ao local desejado. Os estacionamentos dito fora das vias públicas podem ser definidos como lotes e garagens, podendo estes serem públicos ou privados, gratuitos ou pagos. Nas vias públicas, os estacionamentos são encontrados ao longo do meio-fio, podendo ser livres ou pagos, oferecendo um acesso mais fácil e também econômico na maioria dos casos (PEDOT, 2010).

Os estacionamentos de automóveis são um dos maiores problemas encontrados em áreas urbanas, tanto em países desenvolvidos quanto em desenvolvimento. O sistema de estacionamento desempenha um papel fundamental no sistema de tráfego das cidades, a falta dele ocasiona congestionamentos, acidentes de trânsito e poluição ambiental. A cidade que possui um sistema de estacionamento de trânsito bem planejado e eficiente, trás consigo a melhora no transporte urbano, além de elevar a qualidade de vida da população (SEM; AHMED; DAS, 2016, tradução nossa).

Para que a utilização dos meios de transportes individuais se torne eficientes, surge a necessidade da implantação de estacionamentos em áreas públicas e também em áreas privadas. Para a construção desses estacionamentos, é necessário que seja realizado um estudo e um levantamento da demanda atual e revisão futura para identificar a oferta necessária (MELO, 2012).

Atualmente existem algumas tecnologias para o problema da organização de estacionamentos, sendo estas conhecidas como estacionamentos inteligentes. As tecnologias capazes de mostrar aos motoristas vagas livres através de lâmpadas de diferentes cores, e indicam qual o caminho da vaga mais próxima. Existem também tecnologias avançadas ao redor do mundo, estacionamentos autônomos em edificações, ilustrada na figura 3, onde o motorista deixa seu veículo em uma plataforma e um sistema automatizado estaciona o veículo, para retirar o automóvel o motorista precisa fazer uma requisição em uma central.

Figura 3 – Estacionamento vertical automatizado



Fonte: BlastingNews (2015).

No Brasil, o primeiro estacionamento inteligente implantado foi em São Paulo, no Shopping Morumbi. Ilustrado na figura 4, o estacionamento utiliza sensores e LEDs verde - para indicar quais vagas estão disponíveis e vermelhos - indicam quais vagas estão ocupadas, que tem como objetivo diminuir o tempo perdido em encontrar uma vaga para estacionar dentro do shopping. Além de proporcionar mais agilidade ao motorista, este tipo de sistema também ajuda a

administração do shopping a fazer um mapa de ocupação e o controle de fluxo de veículos.

Figura 4 – Estacionamento inteligente shopping Morumbi



Fonte: Estadão(2010).

As vagas disponíveis neste estacionamento (figura 4) são mostradas em painéis indicativos para que a localização de vagas possa ser feita de longa distância, logo na entrada e também em demais setores. Estes painéis fazem a contagem de vagas disponíveis em cada setor e guiam o motorista até elas.

#### 2.1.4 Estacionamento UNESC

Segundo UNESC (2019), atualmente o campus da universidade oferece 2.200 vagas, sendo destas, 34 reservadas aos professores. Segundo CENTAC (2019) a população de acadêmicos é de 9.420, sendo estes matriculados em três períodos, o período matutino possui 2.246, vespertino 1.019 e noturno 6.155. A população de professores contém o total de 755.

Após conversas com funcionários que fazem a segurança do estacionamento, relataram que no período matutino a pouco fluxo de veículos, o período vespertino existe um pequeno crescimento no fluxo do estacionamento, já no período noturno o fluxo é bem mais intenso. Muito dos acadêmicos do período noturno possuem uma rotina de trabalho durante o dia e realizam seus estudos no período noturno, estes optam por muita das vezes o deslocamento através do meio de locomoção individual por ser uma opção mais ágil.

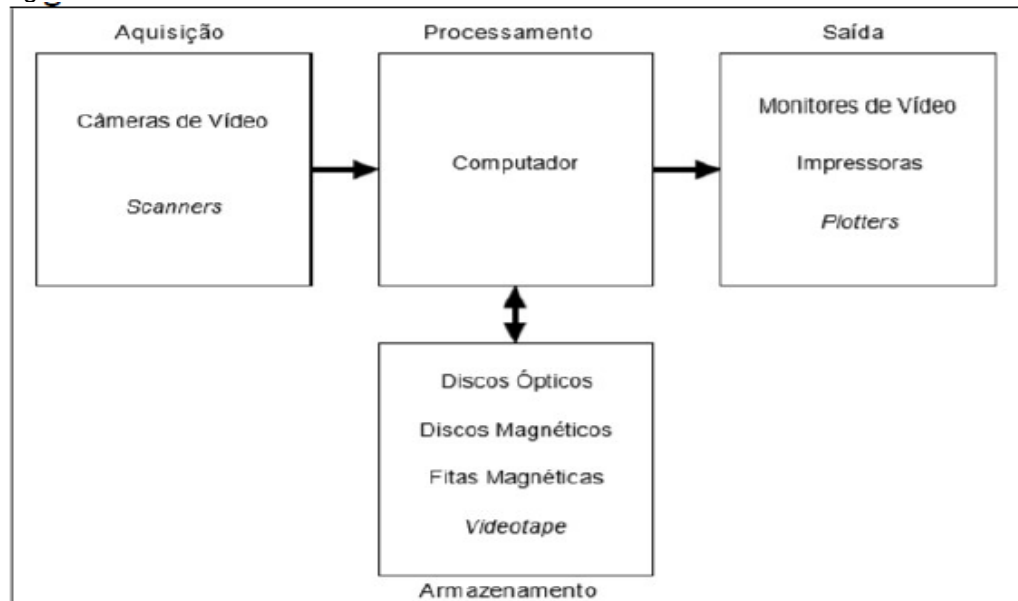
Porém esta opção acaba ocasionando alguns engarrafamentos e gerando dificuldade para encontrar uma vaga de estacionamento disponível devido ao grande fluxo, ocasionando em atrasos no início das aulas.

## 2.2 PROCESSAMENTO DE IMAGENS

O Processamento de Imagens (PID) é a tecnologia que realiza o tratamento de imagens através de algoritmos computacionais, podendo ser encontrada em sistemas inteligentes ou robóticos, sistemas de detecção de objetos, entre outros. Uma imagem digital é composta por vários *pixels*, sendo milhares de pontos que combinam as cores, vermelho, verde e azul para formar as cores do mundo real. O PID tem como seu principal objetivo auxiliar humanos na obtenção de imagens de alta qualidade ou características da imagem original (ZHOU; WU; ZHANG, 2010, tradução nossa).

Com o crescimento da área de PID tornou viável o desenvolvimento de aplicações de análise automática por computador de informações de uma cena, podendo ser definida como “análise de imagens”, “reconhecimento de padrões” ou “visão computacional”. Um sistema de processamento digital de imagem é composto por alguns processos que se aplicam em sistemas de baixo custo até estações de trabalho sofisticadas. Podemos citar estes processos como: aquisição, armazenamento, processamento e exibição. A figura 5 ilustra a sequência destes processos (GONZALEZ; WOODS, 2000).

Figura 5 – Fluxo dos elementos de um sistema PID



Fonte: Marque Filho, Viera Neto (1999).

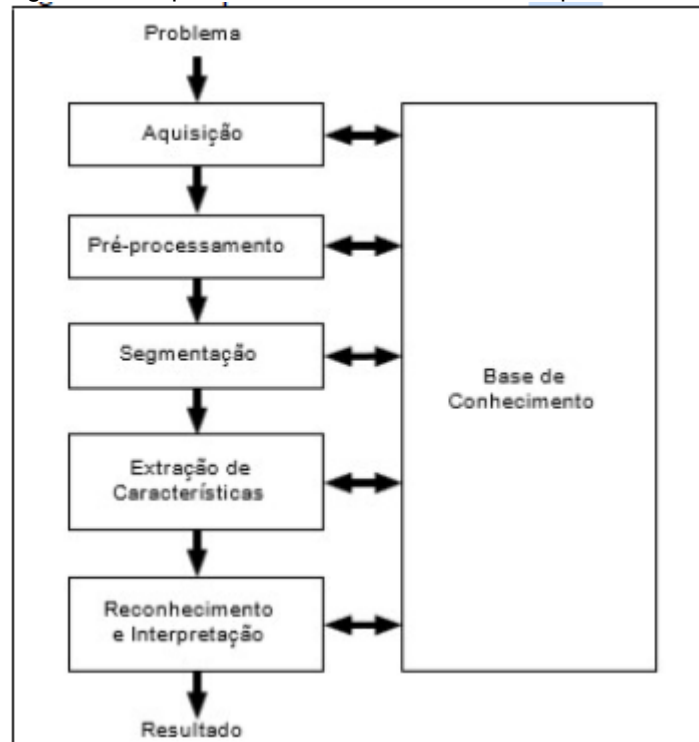
### 2.2.1 Visão Computacional

Visão computacional pode ser observada como a ciência que pesquisa tecnologias que concede as máquinas à capacidade de extrair e processar informações através de imagens (BRUNELLI, 2009, tradução nossa). Considerada como um amplo e complexo campo de estudo, a inteligência computacional, tem sido uma área de pesquisa bastante ativa e crescente nas últimas décadas (MILLER 2011, tradução nossa).

No ponto de vista da engenharia é o processo de automatizar as tarefas que o sistema visual humano desempenha, assim como um médico pode identificar um tumor em uma tomografia ou uma pessoa reconhecer o próprio rosto em uma fotografia (SZELISKI, 2010, tradução nossa).

A primeira etapa executada pela visão computacional se inicia na aquisição, ilustrada na figura 6, sendo está responsável em converter a imagem em uma representação numérica em que o computador possa interpretar. Esta aquisição é composta por dois elementos, o primeiro é um dispositivo sensível a uma faixa de energia no espectro eletromagnético (visível ou invisível aos humanos) que produz um sinal elétrico analógico e o segundo um digitalizador que converte este sinal em uma informação binária. Este processo é executado através de uma câmera, ou outro equipamento eletrônico capaz de capturar imagens (MARQUES FILHO; VIEIRA NETO, 1999).

Figura 6 – Etapas de um sistema de visão computacional



Fonte: Marque Filho, Viera Neto (1999).

O pré-processamento é responsável por melhorar a imagem com o domínio do problema para obter um melhor resultado nas etapas seguintes, por exemplo, a aplicação de contraste ou isolamento de regiões (GONZALEZ; WOODS, 2000).

Na etapa de segmentação o seu objetivo é dividir a imagem em diferentes regiões de acordo com o problema, estas serão analisadas por algoritmos específicos para a extração de informação. Podemos citar como exemplo em uma imagem monocromática (preto e branco), aplicar uma segmentação considerando os seus níveis de cinza ou suas diferenças (GONZALEZ; WOODS, 2000).

A representação e descrição são obtidas através dos resultados da segmentação em uma forma adequada para o processamento computacional que será realizado. Nesta etapa também ocorre uma extração de informações ou características de interesse (MARQUES FILHO; VIEIRA NETO, 1999).

A próxima etapa é composta por dois estágios, sendo o primeiro chamado de reconhecimento e a segunda de interpretação. Esta etapa é responsável por classificar os objetos a partir das informações extraídas da imagem juntamente com

a base de conhecimento já estabelecida anteriormente e também atribuição de um significado a eles (GONZALEZ; WOODS, 2000).

A etapa com maior complexidade é base de conhecimento e também a mais importante, sua função é realizar a comunicação entre os módulos durante os processos e também guia-los (MARQUES FILHO; VIEIRA NETO, 1999).

O resultado final pode ser obtido em qualquer etapa ilustrada na figura 6, irá depender da complexidade de iterações da aplicação. Em alguns casos nem todas as etapas são utilizadas. A última e não menos importante é obter a saída, podendo está ser um conjunto de informações ou até mesmo uma imagem (GONZALEZ; WOODS, 2000).

### 2.2.2 Imagens digitais

A representação visual de um objeto recebe o nome de imagem, sendo está possível ser criada através de aparelhos eletrônicos como, câmeras de vídeo, scanners, dentre outros dispositivos.

Uma imagem digital é uma matriz bidimensional, vista por um computador como um arranjo de elementos em forma de uma matriz, ou seja, uma função  $(x, y)$ , onde  $x$  e  $y$  denotam as coordenadas espaciais (largura e altura) (MARQUES FILHO; VIEIRA NETO, 1999).

A imagem digital é composta por milhares de elementos pequenos denominados *pixels* (pontos na imagem), quanto mais *pixels* possui uma imagem maior é a sua qualidade. Os pontos na imagem não possuem uma medida em centímetros, estes podem obter diferentes tamanhos, ou seja, seu tamanho está diretamente ligado à resolução da imagem (SOARES, 2010).

### 2.2.3 Formatos de arquivos

Existem dois modos diferentes para a criação, armazenamento e processamento de imagens digitais, uma delas é através de bitmaps (mapa de bits), conhecida também como *pixel maps* (mapa de pixels) ou *raster* (varredura), e o segundo formato, por vetores (MARQUES FILHO; VIEIRA NETO, 1999).

Os gráficos *bitmaps* são os modos mais comuns, as imagens possuem variações complexas em suas formas e cores, podemos citar como exemplo, nos

frames de um vídeo, a desvantagem de trabalhar com arquivo tipo *raster*, é que ao ampliá-lo, ocorre a perda de definição e uma diminuição considerável na qualidade. O formato definido por vetores é a mais utilizada quando há predominância de linhas e preenchimento simples, por exemplo, em diagramas e desenhos simples (MARQUES FILHO; VIEIRA NETO, 1999).

#### **2.2.4 Espaço cor**

O método de espaço cor é utilizado para especificar, criar e visualizar cores. Para interpretar a cor de um ponto de uma determinada imagem são necessárias três informações. O cérebro humano identifica as cores através do brilho (percepção de mais ou menos luz em um ponto), a matriz (percepção de qual cor visível, vermelho, amarelo, verde ou azul), saturação (percepção de intensidade que um ponto da imagem apresenta sua matriz). Na computação, uma cor pode ser representada pela quantidade de emissão de luz vermelha, verde e azul (RGB) necessária para obter a percepção da cor desejada (SZELISKI, 2010, tradução nossa).

O modelo RGB é mais comumente utilizado na computação. É baseado em um sistema de coordenadas cartesianas, podendo ser comparado para análise como um cubo, onde três de seus vértices são cores primárias, e os demais são secundárias, o vértice junto à origem é o preto e o mais afastado é o branco como ilustra a figura 7 (MARQUES FILHO; VIEIRA NETO, 1999).



da imagem e possuem uma alta frequência na imagem, são chamados de passa-baixa. Resultando este em efeitos de borrado, pois elementos de alta frequência, que são as bordas e pequenos detalhes são eliminados. Atuando de forma contrária temos os filtros de passa-alta, estes realçam as bordas e os pequenos detalhes. O ultimo e mais complexo filtro chamado de passa-faixa, podendo remover ou atenuar partes da imagem (MARQUES FILHO; VIEIRA NETO, 1999).

Um exemplo de aplicação de filtros ilustrado na figura 8, selecionando um ponto central e aplicando o filtro da média, e sendo alterado a partir da média dos valores de seus *pixels* vizinhos. O objetivo deste filtro é a remoção de ruídos, porém preservando bordas e detalhes finos da imagem.

Figura 8 – Efeito do filtro da média em uma imagem

10	10	0	0	0	0	0	0	20	20	10
90	80	80	70	65	65	40	40	30	30	20
35	35	35	35	50	50	42	42	30	30	20
90	180	180	180	180	180	120	100	90	70	50
200	200	190	190	180	180	120	120	90	72	55
200	200	190	190	150	150	120	120	90	72	50
205	210	190	190	150	150	120	120	90	72	50
204	210	205	200	150	150	120	120	90	73	50
210	210	210	200	150	150	120	120	80	73	40
190	190	190	185	150	150	120	120	80	69	23
190	190	190	185	145	145	115	115	80	69	23

Fonte: Mexas (2014).

### 2.3 OPENCV

A biblioteca *Open Source Vision Library* (OpenCV), desenvolvida pela Intel em 2000, é um conjunto de mais de 2.500 algoritmos de programação em código aberto, voltada para o desenvolvimento de aplicações em visão computacional. Proporciona utilidades como, manipulação de imagens, desenvolvimento de algoritmos de processamento de imagens, análise de movimentos, reconhecimento de objetos, entre outros (CUNHA, 2013).

Desenvolvida nas linguagens C e C++, sendo compatível com os principais sistemas operacionais, Linux, Windows e Mac OS X. Projetada com o objetivo de tornar a visão computacional mais aberto a usuários e também

desenvolvedores nas áreas de comunicação entre humano e computador em tempo real (BRADSKI; KAEHLER, 2008, tradução nossa).

A construção do OpenCV surgiu durante uma pesquisa da Intel para aplicações de intenso uso da CPU. Um dos pesquisadores da Intel notou que nas universidades como o MIT possuíam grupos de alunos com uma estrutura de visão computacional bem desenvolvida, e os alunos davam continuidade aos seus projetos reaproveitando o código já existente e os aperfeiçoando. A partir deste método, o OpenCV foi criado para tornar esta estrutura de visão computacional universalmente disponível (BRADSKI; KAEHLER, 2008, tradução nossa).

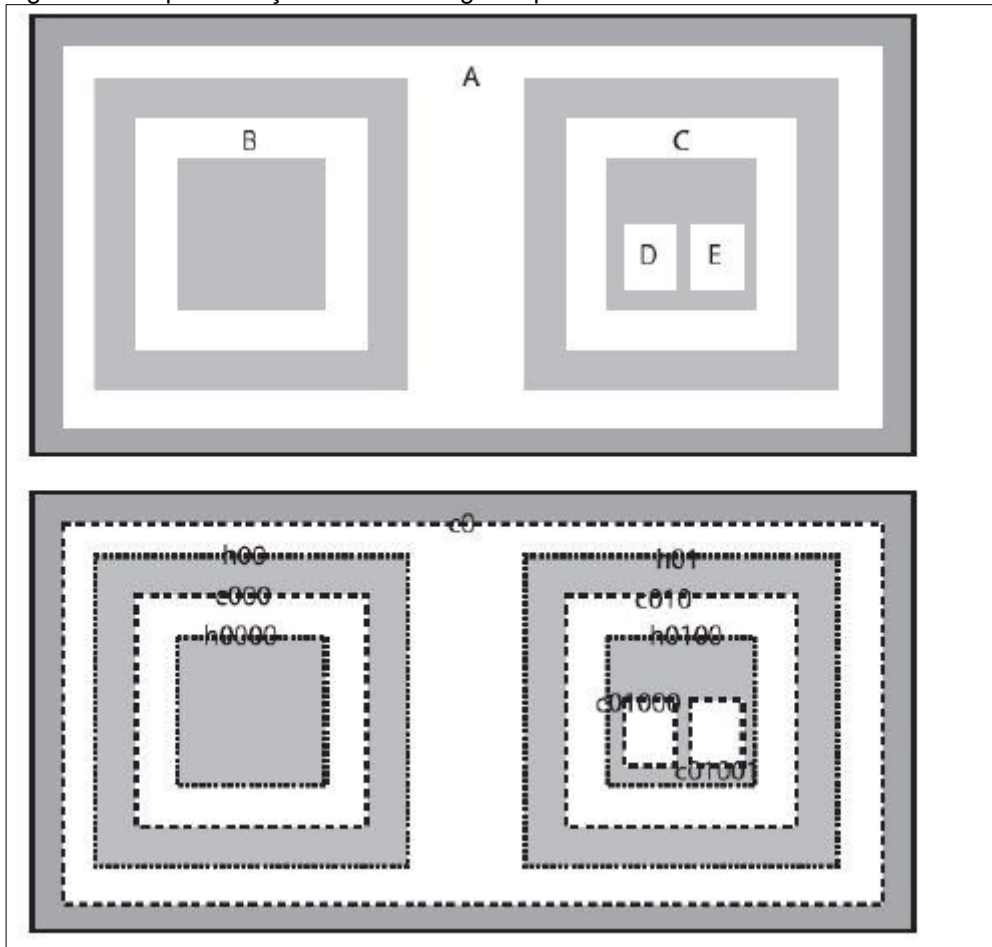
### 2.3.1 Contornos

Em processamento de imagens a definição de contorno pode ser dada como uma curva que liga todos os pontos contínuos, com mesma cor e intensidade é de grande utilidade na análise de formas e detecção e reconhecimento de objetos.

A biblioteca OpenCV possui duas funções para aplicação de contornos em imagens, são elas: `cv2.findContours` e `cv2.drawContours`. A função `cv2.findContours` quando aplicada é capaz de detectar os diferentes segmentos da imagem através de suas bordas e contornos. Já a `cv2.drawContours`, é responsável por desenhar os contornos e também pode ser utilizada para desenhar qualquer forma, através de métodos aplicados utilizando a função em questão (ROSEBROCK, 2016, tradução nossa).

A figura 9 exemplifica o método `cv2.findContours` em ação. Na parte superior do exemplo temos uma imagem que contém regiões brancas em um fundo escuro. Na parte inferior da ilustração temos a mesma imagem, com os contornos que o método irá encontrar. Os contornos podem ser rotulados com `cX` ou `hX`, onde a letra `c` significa *contour*, ou seja, contorno, e `h` significa *hole*, ou seja, buraco, e `X` é um número (BRADSKI; KAEHLER, 2008, tradução nossa).

Figura 9 – Representação de uma imagem após utilizado o método : *cvFindContours()*



Fonte: Bradski e Kaehler (2008).

### 2.3.2 Laplace

O propósito da utilização do método de Laplace é a aplicação de bordas para determinar os contornos dos objetos presentes na imagem. Esta função pertence ao conjunto de algoritmos que compõem a biblioteca OpenCV através do comando *cv.Laplacian(src, dst, ddepth, ksize)*, onde *src* e *dst* são as imagens de origem e destino, *ddepth* é um código que indica a quantização da imagem resultante e *ksize* é o tamanho da máscara (GONZALEZ, WOODS, 2007).

O método torna-se eficiente devido à definição do próprio operador Laplaciano, que é utilizado para calcular a segunda derivada da imagem. Diferentemente dos filtros de primeira ordem que detectam as arestas com base no máximo e mínimo, o Laplacian detecta as arestas em cruzamento em zero, ou seja, onde a alteração de valor negativo para positivo ou vice-versa. A aplicação do método obtém-se como resultado o destaque de regiões de uma imagem que

contém mudanças rápidas de intensidade, assim como outros operadores matemáticos como *Sobel* e *Scharr*, o Laplace é frequentemente utilizado para a detecção de bordas (ROSEBROCK, 2016, tradução nossa).

Figura 10 – Representação matemática do método de Laplace

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

Fonte: Singh e Kang (2019).

A utilização de filtros juntamente como o método (figura 9) é necessária, estes aplicam a suavização dos ruídos ou diferenças indesejáveis de gradiente devido à maior sensibilidade ajustada de acordo com o tamanho do *Kernel*, sendo definido entre um e cinco, quanto maior, maior será a sensibilidade de detecção de bordas devido ao maior processamento de *pixels* (SINGH, KANG, 2019, tradução nossa).

### 2.3.3 YAML

O formato de arquivo chamado *Yet Another Markup Language* (YAML) é um tipo de serialização de dados legível por humanos, o conceito deste tipo de arquivo foi inspirado em linguagens de programação como o XML, C, Python, Perl. A leitura do YAML é mais simples quando comparada ao XML, sua estrutura possibilita representar tudo o que pode ser escrito em um arquivo XML, porém de uma forma mais compactada e limpa das estruturas de dados (FONSECA, SIMÕES, 2007).

A figura 11 ilustra a estrutura do arquivo YAML e XML.

Figura 11 – Estrutura de um arquivo YAML e XML

<pre> players:   Vladimir Kramnik: &amp;kramnik     rating: 2700     status: GM   Deep Fritz: &amp;fritz     rating: 2700     status: Computer   David Mertz: &amp;mertz     rating: 1400     status: Amateur  matches: -   Date: 2002-10-04   White: *fritz   Black: *kramnik   Result: Draw -   Date: 2002-10-06   White: *kramnik   Black: *fritz   Result: White </pre>	<pre> &lt;?xml version="1.0"?&gt; &lt;club&gt;   &lt;players&gt;     &lt;player id="kramnik"       name="Vladimir Kramnik"       rating="2700"       status="GM" /&gt;     &lt;player id="fritz"       name="Deep Fritz"       rating="2700"       status="Computer" /&gt;     &lt;player id="mertz"       name="David Mertz"       rating="1400"       status="Amateur" /&gt;   &lt;/players&gt;   &lt;matches&gt;     &lt;match&gt;       &lt;Date&gt;2002-10-04&lt;/Date&gt;       &lt;White refid="fritz" /&gt;       &lt;Black refid="kramnik" /&gt;       &lt;Result&gt;Draw&lt;/Result&gt;     &lt;/match&gt;     &lt;match&gt;       &lt;Date&gt;2002-10-06&lt;/Date&gt;       &lt;White refid="kramnik" /&gt;       &lt;Black refid="fritz" /&gt;       &lt;Result&gt;White&lt;/Result&gt;     &lt;/match&gt;   &lt;/matches&gt; &lt;/club&gt; </pre>
---	--

Fonte: Fonseca e Simões (2007).

## 2.4 COMUNICAÇÃO DE DADOS

A comunicação de dados trata da transmissão de sinais e troca de informações através de um ou mais meios físicos, de uma forma confiável e eficiente. Para que esta comunicação seja possível os meios não precisam ser iguais, porém para que este processo ocorra, os dispositivos devem possuir hardware e software integrados (PERCÍLIA, 2019).

Segundo Percília (2019), a eficiência de um sistema de comunicação de dados depende fundamentalmente de três características:

- a) entrega (delivery): o envio deve ser efetuado somente para o dispositivo de destino;
- b) confiabilidade: a entrega e funcionamento dos dados deve ser garantida pelo sistema;

- c) tempo de atraso: a transmissão dos dados deve ser eficiente, tendo está um tempo predeterminado para o envio dos dados.

Segundo Pereira (2012) um sistema de comunicação de dados é constituído por cinco elementos.

- a) mensagem: dados a serem transmitidos, constituído por números, texto, imagens, vídeos, áudios;
- b) transmissor: dispositivo responsável pelo envio da mensagem;
- c) receptor: dispositivo responsável por receber a mensagem;
- d) meio: caminho físico por onde viaja a mensagem originada e dirigida ao receptor;
- e) protocolo: conjunto de regras que governa o envio e recebimento dos dados.

#### **2.4.1 Protocolos de comunicação**

Para que ocorra uma comunicação de dados entre dispositivos seja eficiente e sem erros existem os protocolos de comunicação de dados, definidos como um conjunto de regras e procedimentos com objetivo de padronizar a comunicação.

Para Castelluci (2011), é essencial um conjunto de regras estabelecido para comunicação entre dispositivos, pois os diversos dispositivos nem sempre se comunicam através da mesma linguagem.

##### **2.4.1.1 TCP**

As características do protocolo *Transmission Control Protocol* (TCP) é fazer a circulação simultânea de informações de fontes distintas e permitir o início e o fim de uma comunicação de maneira correta (PILLOU, 2017).

O funcionamento do protocolo TCP é baseado em conexões, quando um dispositivo inicia uma comunicação a uma determinada página WEB, por exemplo, este se denomina como cliente, este solicita a outro dispositivo que contem a informação desejada, chamado de servidor, após o processo ser concluído, o cliente faz uma checagem de erros nos pacotes por meio de um sistema responsável por detectar fragmentos corrompidos ou desordenados. Se ao fim da comunicação a

informação esteja correta, o cliente confirma o recebimento completo e sem erros, caso contrário, o servidor realiza o envio novamente até que o cliente confirme que os dados estão corretos (GUGELMIN, 2014).

Para que a comunicação seja efetuada entre dois ou mais dispositivos é necessário um acordo entre eles, este acordo é chamado de *handshake* de três vias, responsável por verificar se ambas as partes estão prontas para o início da transmissão. Após a checagem é possível que as estações transfiram blocos de dados entre si, estes blocos de tamanho variável são organizados por informações (STALLINGS; LAWRIE, 2013).

## 2.5 CÂMERAS IP

A utilização de câmeras de monitoramento se iniciou no ano de 1965, onde a polícia de Nova York instalou câmeras em áreas públicas, as chamadas câmeras analógicas, como não possuíam um sistema de armazenamento de imagens, precisavam ser constantemente monitoradas. Em 1970, se iniciou a gravação de imagens em cassetes de vídeo, com esta nova funcionalidade o monitoramento com câmeras ganhou popularidade em todo o mundo e sendo aplicado para o monitoramento de grandes lojas, sistema rodoviário e bombas de gasolina (KIRCH, 2009).

Nos anos 80, surgiram as câmeras CCD, inserido um micro-chip ao sistema sendo possível captar imagens em locais com baixa luminosidade e até durante a noite, esta nova funcionalidade permitiu que as câmeras pudessem ser utilizadas de forma mais abrangente e efetiva. Os sistemas de monitoramento foram ganhando novas funcionalidades de acordo com o avanço da tecnologia, foi a partir dos anos 90 que a possibilidade do armazenamento de imagens de várias câmeras simultâneas em um só sistema. A evolução dos computadores também contribuiu para estes sistemas, sendo possível inserir disco rígido aos sistemas e alterando todo o processo de monitoramento e ao acesso as imagens (KIRCH, 2009).

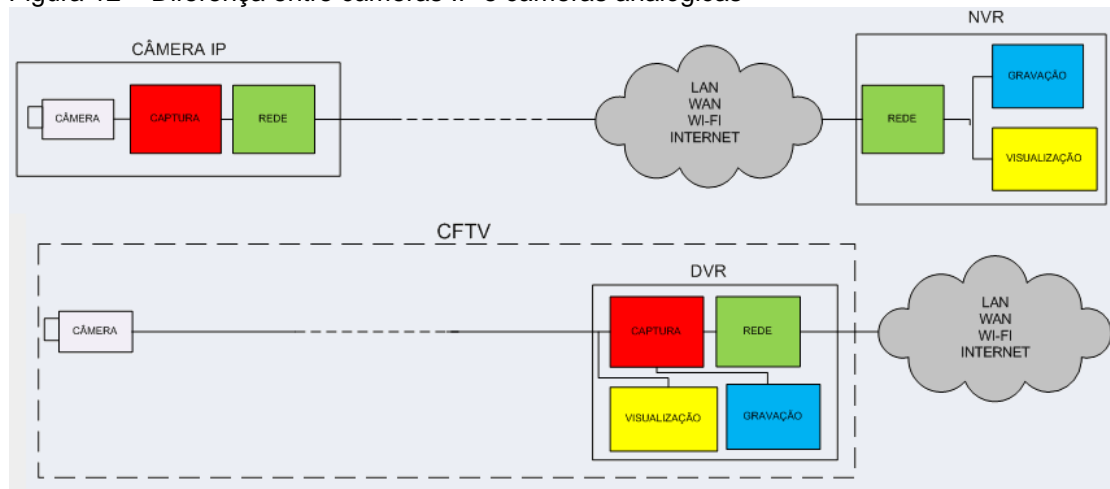
Uma câmera de monitoramento possui basicamente os seguintes componentes, um sensor de imagem, circuito de análise e processamento de vídeo, interface de rede, integrados no mesmo equipamento. A composição interna de uma câmera é formada por uma complexa estrutura de processos, gerenciadas por um

sistema operacional próprio, sendo este o responsável por interligar todas as funções necessárias para o seu funcionamento (SAMBUGARO, 2016).

Com o grande avanço da tecnologia e a grande demanda pela procura de um sistema de monitoramento eficiente, surgiram às câmeras IP, sendo estas podendo ser acessadas por longa distância via browser ou aplicativos em dispositivos móveis (SAMBUGARO, 2016).

As câmeras IP surgiram no mercado de vigilância eletrônica em 2006, desenvolvida pela empresa Axis Communications, utilizando a plataforma Linux embarcada. A integração e criação de programas e sistemas de gerenciamento por parte de terceiros se iniciou após a própria Axis disponibilizar a sua API. Esta iniciativa da empresa resultou em um grande crescimento no mercado de vigilância e também na criação de sistemas personalizados para determinadas aplicações e propósitos (KIRCH, 2009).

Figura 12 – Diferença entre câmeras IP e câmeras analógicas



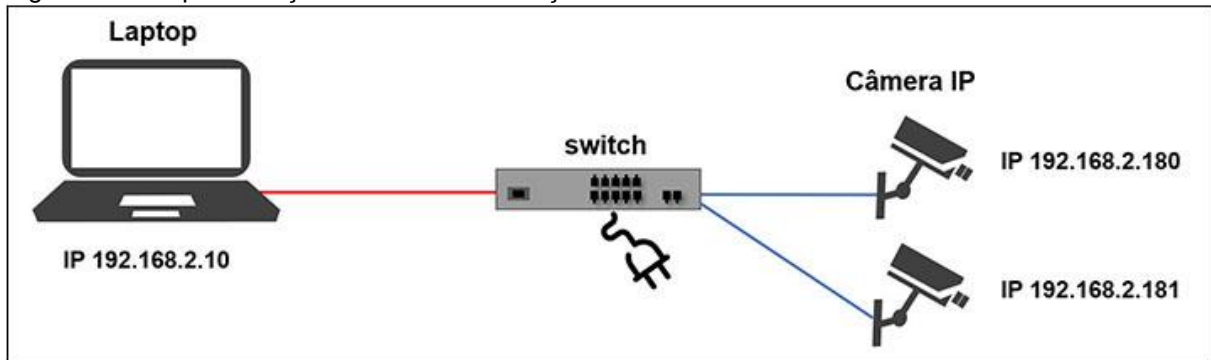
Fonte: Almeida (2015).

### 2.5.1 Comunicação câmera IP

A comunicação com a câmera IP (figura 13) funciona exatamente como um computador ou qualquer outro equipamento com dispositivo de rede conectado a uma rede residencial ou corporativa. Ao ser conectado a rede o equipamento recebe um endereço de IP, sendo este distribuído por servidores ou outro aparelho que realize o gerenciamento da rede, o endereço pode ser definido manualmente ou automaticamente irá depender das definições de rede do local. Após definição do

endereço IP ao equipamento a comunicação com os demais aparelhos eletrônicos presente na mesma rede é estabelecida (ALMEIDA, 2015).

Figura 13 – Representação de uma comunicação com câmeras IP



Fonte: Martins (2018).

### 2.5.2 Infravermelho

O infravermelho é um tipo de radiação eletromagnética que apresenta frequência menor que a luz vermelha e, por isso, não é visível ao olho humano. O funcionamento do infravermelho é semelhante a uma lanterna, em um ambiente sem luzes, onde o olho humano não é capaz de enxergar, ligamos a mesma para poder visualizar o ambiente. Em câmeras de monitoramento, por sua vez, é ativado para aumentar a sua capacidade de captação de imagens (GREATEK, 2017).

O infravermelho é considerado como um componente extremamente importante nas câmeras de monitoramento, pois será ele que irá definir o quanto a câmera irá capturar em um ambiente sem iluminação, ou em locais com baixa luminosidade. Quanto mais forte for o infravermelho, maior será a iluminação do ambiente e mais distante ela se propagará (GREATEK, 2017).

### 2.5.3 Resolução

A resolução de imagem das câmeras IP é outro elemento que deve ser observado na aplicação de monitoramento em determinada área, este é outro elemento forte que compõem além dos citados anteriormente, disponíveis em resoluções de 1 *megapixel* (HD), 2 *megapixel* (Full HD), 3 *megapixel* e 4 *megapixel*. Em alguns modelos podemos encontrar a tecnologia IR inteligente, o objetivo desta tecnologia é manter a nitidez do objeto observado esteja muito perto da lente (LUIZ, 2016).

## 2.6 DISPOSITIVOS MÓVEIS

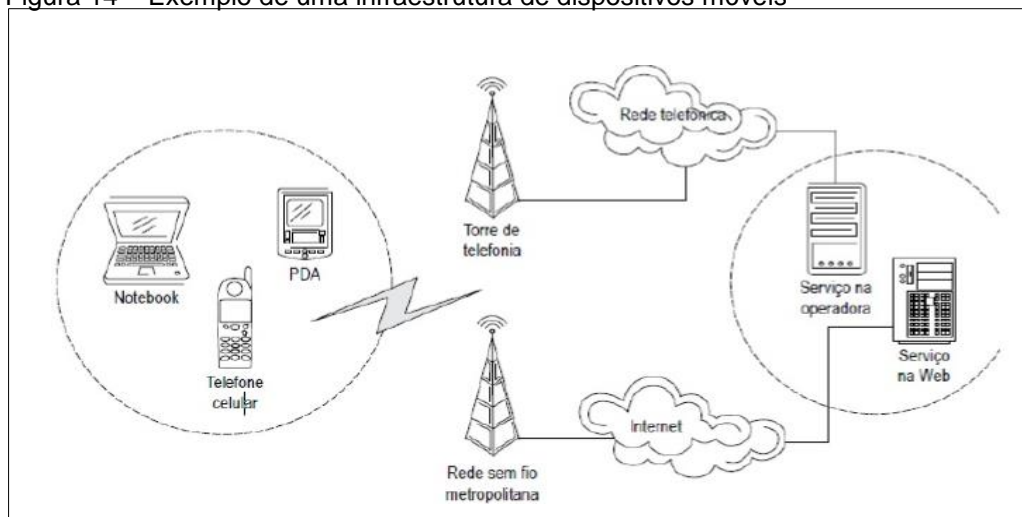
O crescimento no mercado de *smatphones* obteve um ritmo elevado nos últimos anos, no Brasil e no mundo. Em agosto de 2018, o Brasil chegou à marca de 234,7 milhões de telefones celulares, chegando a densidade de 111,96 celulares a cada 100 habitantes, o número de celulares já é maior que a quantidade de habitantes (TELECO, 2018).

Um dos principais motivos que influencia o alto crescimento está na busca de novas tecnologias e constantes melhorias, voltadas a parte de hardware e sistemas operacionais que compõem estes dispositivos.

A criação de plataformas ou sistemas operacionais móveis foi desenvolvida com o objetivo de disponibilizar uma melhor comunicação entre usuários e *smartphones*, tendo como finalidade manipular os recursos dos dispositivos, como: câmera, lista telefônica, entre outros (MAZUCHETTI, 2015).

Os dispositivos móveis possibilitam aos usuários acesso a serviços independentes de sua localização. Para que o acesso seja possível é necessário que haja uma infraestrutura de comunicação sem fio, como ilustrado da figura 14, onde mostra a infraestrutura de redes de telefonia celular e as redes sem fio que são as necessárias para os acessos aos serviços móveis (JOHNSON, 2007).

Figura 14 – Exemplo de uma infraestrutura de dispositivos móveis



Fonte: Johnson (2007).

## 2.7 NAVEGADORES

Com o avanço da Internet, a quantidade de informações na rede também cresceu e, para usar estes recursos disponíveis na rede, é necessário um software que possibilite a busca por elas. São estes chamados de navegadores ou *web browser* (ORGERA, 2019, tradução nossa).

Em 1990, o cientista Tim Berners-Lee criou o primeiro navegador chamado WWW abreviação de *WorldWideWeb*, a principal função deste novo sistema era interligar as universidades para que os trabalhos e pesquisas acadêmicas fossem utilizadas e disponibilizadas com agilidade e gerando a contribuição de informações dos núcleos envolvidos. Foi este mesmo cientista que criou o protocolo *Hyper Text Transfer Protocol* (HTTP) e a linguagem de programação *HyperText Markup Language* (HTML), ferramentas essenciais para o funcionamento dos navegadores (BARWINSKI, 2009).

Na atualidade os navegadores web são indispensáveis quando se trata de navegar na rede ou realizar alguma operação que facilite o dia-a-dia, já que quase tudo é possível através da internet como, pagar contas, conhecer pessoas, realizar compras, entre outras. Os navegadores mais conhecidos são Google Chrome, Mozilla Firefox, Ópera, cada um destes navegadores possui a sua própria funcionalidade, compatibilidade, interface, segurança e outras variáveis (OLIVEIRA, 2014).

Os navegadores web modernos possuem um poderoso conjunto de ferramentas são estas responsáveis por inspecionar e interpretar arquivos HTML, CSS e JavaScript, estes arquivos são carregados no momento em que o usuário requisita uma pesquisa em determinada página da web (NESBITT, 2018, tradução nossa).

### 2.7.1 WEBAPPS

Aplicações desenvolvidas com tecnologias utilizadas geralmente no desenvolvimento de sites, como: HTML5, CSS e *JavaScript*, além de PHP e Java, estas compõem os chamados *Webapps* ou *mobile web sites* (WILKEN, 2016, tradução nossa).

Os chamados *Webapps* têm como principais vantagens, a compatibilidade e portabilidade com a maioria dos dispositivos móveis independente de sua plataforma, possibilitando o acesso diretamente de um navegador web, com isso não tendo a necessidade de instalação. Grande parte destas aplicações são sites comuns da internet, mas com o objetivo de se adequarem a dispositivos menores, por meio de tecnologias responsivas que se adaptam as mais variadas resoluções (WILKEN, 2016, tradução nossa).

### 3 TRABALHOS CORRELATOS

Para o desenvolvimento deste projeto de pesquisa, foram pesquisados alguns trabalhos correlatos para validar a viabilidade do mesmo.

#### 3.1 EM RUMO A UM SISTEMA AUTOMÁTICO DE CONTROLE DE ACESSO DE VEÍCULOS AUTOMOTIVOS: RECONHECIMENTO DE CARACTERES EM PLACAS DE VEÍCULOS

Trabalho de conclusão de curso desenvolvido por Daniel Rocha Gualberto apresentado na Universidade Federal de Ouro Preto em 2010, este trabalho de pesquisa aborda os conceitos de um Sistema Automático de Controle de Acesso de Veículos Automotivos (SACAVA), destacando o sistema de Reconhecimento de Caracteres de Placas de Identificação de Veículos (RCPIV).

A primeira solução deste tipo de sistema foi criado em 1976 no Reino Unido, com o objetivo de identificar carros roubados e apenas em 1981 ocorreu à primeira identificação.

O objetivo geral da pesquisa foi reconhecer os caracteres de placas de licença de veículos a partir de imagens digitais, apenas em modo de simulação, não havendo interação direta com o ambiente real.

Para o desenvolvimento deste trabalho foi necessário utilizar diferentes técnicas para localização da placa, pré-processamento, segmentação e reconhecimento dos caracteres. Para obtenção de resultados foram usadas 343 fotos de placas veiculares em diferente estados como placas sujas, amassadas e com reboque na frente, tendo como resultado positivo o reconhecimento de 75%.

#### 3.2 ESTACIONAMENTO INTELIGENTE

Trabalho de conclusão de curso desenvolvido por Said Hikmat Abd Alhak apresentado no Centro Universitário de Brasília em 2011, em seu trabalho de conclusão de curso o objetivo foi desenvolver um protótipo de estacionamento inteligente a fim de mostrar aos motoristas a quantidade de vagas livres e o local e em que setores elas se encontram, chegando ao setor, outro display guia o motorista até a vaga.

A detecção de vaga é feita por um sensor de presença (transmissor e receptor infravermelho), e cada vaga possui dois LEDs indicativos, sendo um verde que indica a vaga disponível, e outro vermelho que indica que a vaga está ocupada, todas essas informações são enviadas para um microcontrolador **PIC**. A quantidade de vagas disponíveis será apresentada através de *displays*, facilitando o fluxo de carros e diminuindo o tempo gasto em filas.

Para o desenvolvimento deste protótipo foi preciso obter conhecimento em Circuitos Eletrônicos, Microprocessadores e Microcontroladores. Os resultados foram satisfatórios obtendo informações sobre histórico das vagas e suas estatísticas, como o tempo de ocupação, horários de maior fluxo e as vagas mais e menos ocupadas.

### 3.3 RECONHECIMENTO AUTOMÁTICO DE PLACAS VEICULARES UTILIZANDO A TECNOLOGIA OCR E A PLATAFORMA RASPBERRY PI APLICADA NA FISCALIZAÇÃO ELETRÔNICA DE RODOVIAS

Trabalho de conclusão de curso desenvolvido por Jackson Guizzo Zanette apresentado na Universidade do Extremo Sul Catarinense de Santa Catarina defendido em 2016, foi desenvolvido um protótipo de um radar de fiscalização eletrônica de rodovias, tendo como objetivo a utilização da tecnologia OCR para reconhecimento de placas e a plataforma RapsBerry Pi para medição de velocidade dos veículos (ZANETTE, 2016).

O desenvolvimento deste protótipo se iniciou através de diferentes técnicas para localização da placa aplicando conceitos de, pré-processamento de imagem, segmentação e reconhecimento dos caracteres, com auxílio das bibliotecas OpenCV e Tesseract.

Para obter resultados neste projeto, foi necessário o desenvolvimento de uma maquete simulando uma estrada com boa iluminação e um carro de brinquedo para detecção da placa e velocidade do protótipo, foram realizadas cerca de 50 passagens com diferentes placas e obtendo apenas 7 falhas de reconhecimento das placas (ZANETTE, 2016).

### 3.4 USO DE VISÃO COMPUTACIONAL EM DISPOSITIVOS MÓVEIS PARA AUXÍLIO À TRAVESSIA DE PEDESTRES COM DEFICIÊNCIA VISUAL

Dissertação de mestrado desenvolvido por Kelly Sousa e Mauricio Marengoni apresentado na Universidade Presbiteriana Mackenzie de São Paulo defendido em 2013. Nesta dissertação de mestrado, desenvolveu-se um aplicativo utilizando a biblioteca OpenCV para o tratamento de imagens e a plataforma Android para auxiliar deficientes visuais na travessia de ruas.

Para a realização deste projeto, foi realizada uma pesquisa sobre a usabilidade deste tipo de aplicação, e constatado que já existem aplicativos voltados para cegos, tais como presença de luz para cegos totais, notas de dinheiro, cores e o reconhecimento de texto impresso.

O aplicativo é ativado através de um toque do usuário, assim, o software faz um reconhecimento da área para detecção da faixa de pedestre, após o reconhecimento da faixa o aplicativo informa o usuário através da vibração do aparelho ou por voz se existe uma faixa de pedestre e se existem pessoas no caminho.

### 3.5 PLANEJAMENTO DE TRAJETÓRIA PARA ESTACIONAMENTO DE VEÍCULOS AUTÔNOMOS

Trabalho de conclusão de curso desenvolvido por Marcos Gomes Prado apresentado no Instituto de Ciências Matemáticas e de Computação de São Paulo defendido em 2013. Teve como objetivo o desenvolvimento de um sistema inteligente para gerar e executar uma trajetória para o estacionamento de veículos autônomos em um ambiente semiestruturado.

O desenvolvimento do sistema foi dividido em três etapas, primeiramente a localização da vaga disponível, o planejamento do caminho até a vaga escolhida e o controle do veículo. A detecção da vaga é feita através de um laser, após a detecção o sistema é responsável pela parte de definição da trajetória até a vaga e também pelo controle do veículo, guiado o mesmo até a vaga através do caminho planejado.

A detecção das vagas é feita por sensores instalados no veículo e estes fazem a comunicação com o sistema, gerando um caminho para uma vaga

disponível e também conduzindo o veículo controlando sua aceleração e rodas até a vaga.

### 3.6 PROCESSO DE RECONHECIMENTO NÃO SUPERVISIONADO DE ÁREAS DE ESTACIONAMENTO

Trabalho de conclusão de curso desenvolvido por Antonio Henrique Mexas apresentado na Universidade Presbiteriana Mackenzie de São Paulo defendido em 2014. O presente projeto consiste na análise de algoritmos referente a identificação e reconhecimento de áreas pavimentadas e sinalizadas de estacionamento, para a modelagem de um algoritmo composto de identificação de áreas de estacionamento.

Para a análise de resultados foram utilizadas 30 imagens de alta resolução obtidas pelo site Satellite Imaging Corporation. Os conceitos utilizados para a identificação foram: busca por veículos estacionados, busca por marcação sem veículos estacionados e busca por áreas de asfalto. Com estes três conceitos unidos obtiveram-se: para identificação de pontos utilizando o método de busca para identificação de carros estacionados uma taxa de acerto média de 98% e com falso positivo médio de 2%. Mas, o objetivo do projeto é a identificação de área de todo o parque de estacionamento e o modelo composto pela identificação de carros estacionados sem veículos teve o melhor aproveitamento com 85% de média de acerto e uma taxa de falso positivo média de 15%.

## **4 CONTROLE DE VAGAS DE ESTACIONAMENTO A PARTIR DE TRATAMENTO DE IMAGENS EM TEMPO REAL COM A BIBLIOTECA OPENCV**

Baseando-se em todo referencial teórico coletado e no conhecimento adquirido a partir dele, o trabalho consiste no desenvolvimento de um protótipo na linguagem de programação *Python* para a identificação de vagas de estacionamento em tempo real, com auxílio da biblioteca OpenCV e uma página *WEB* onde é possível visualização do local das vagas livres e ocupadas, o intuito deste projeto é a criação de algo ágil e útil para o dia-a-dia das pessoas.

### **4.1 METODOLOGIA**

Para o desenvolvimento deste projeto de pesquisa foram realizadas algumas etapas, a primeira etapa exercida foi referente ao levantamento bibliográfico, a qual foi efetuada por meio de publicações, artigos, livros e pesquisas em geral na internet.

Em seguida, iniciou-se a escrita do referencial teórico, descrevendo os principais assuntos que compõem os procedimentos necessários para o desenvolvimento do protótipo proposto.

Na etapa em questão foram abordados temas como: mobilidade urbana e o crescimento desordenado de automóveis nas vias e as consequências resultantes, dando maior ênfase na dificuldade de encontrar uma vaga de estacionamento, sendo ela em locais públicos ou privados; métodos de desenvolvimento utilizando a visão computacional e a biblioteca OpenCV para a identificação de objetos e aplicação de filtros.

As demais etapas são constituídas no desenvolvimento do protótipo, iniciando-se primeiramente pelo levantamento de recursos necessário, em seguida a instalação da câmera de monitoramento no estacionamento a ser aplicado o software, após isso foi dado início a programação para identificação das vagas e objetos nas áreas demarcadas, seguido da configuração do banco de dados e do servidor para visualização dos dados na página *WEB* e por fim os testes de reconhecimento das vagas ocupadas e livres.

#### 4.1.1 Ferramentas e levantamento de requisitos

A etapa de requisitos necessários realizou-se o estudo sobre o conceito de visão computacional, a fim de extrair informações da teoria para, então, aplicá-las no desenvolvimento do software para detecção de vagas de estacionamento. Para melhor entendimento da tecnologia a ser utilizada, foi realizada também uma pesquisa sobre projetos desenvolvidos com o mesmo objetivo, no caso a biblioteca de tratamento de imagens OpenCV. A partir dos conhecimentos obtidos através desta pesquisa foi possível o levantamento dos recursos iniciais para o desenvolvimento do protótipo.

Para o desenvolvimento do protótipo, foi adquirida uma câmera IP com comunicação via protocolo TCP, ilustrada na figura 15. A utilização deste dispositivo teve como finalidade a captura do vídeo, pois o objetivo deste projeto é a realização de tratamento de imagens em tempo real para a detecção das vagas de estacionamento, possibilitando a visualização de vagas ocupadas e livres para os usuários de automóveis e tornando ágil a localização das mesmas.

Figura 15 – Representação de uma câmera IP



Fonte: Do autor.

Com a aquisição da câmera IP se fez necessário à procura de uma área de estacionamento, para definir as coordenadas das vagas e assim ser desenvolvido o software do protótipo.

Após a pesquisa realizada, o método encontrado para definição das vagas de estacionamento se fez através de pontos da imagem e salvos em um arquivo de texto, estes pontos foram gerados com a implementação de métodos da biblioteca OpenCV descritos no capítulo 2. Dentro da área demarcada de cada vaga, como ilustra a figura X, o software de protótipo lê os pontos definidos no arquivo de texto e os apresenta, em seguida realiza a verificação de alteração na escala de cinza em cada uma das vagas, caso haja alteração, será considerado que há um carro na vaga.

Como citado anteriormente, para a realização da captura de imagem para identificação das vagas e carros, foram utilizados alguns equipamentos, aos quais são listados na tabela 1.

Tabela 1 - Equipamentos utilizados para a montagem da simulação

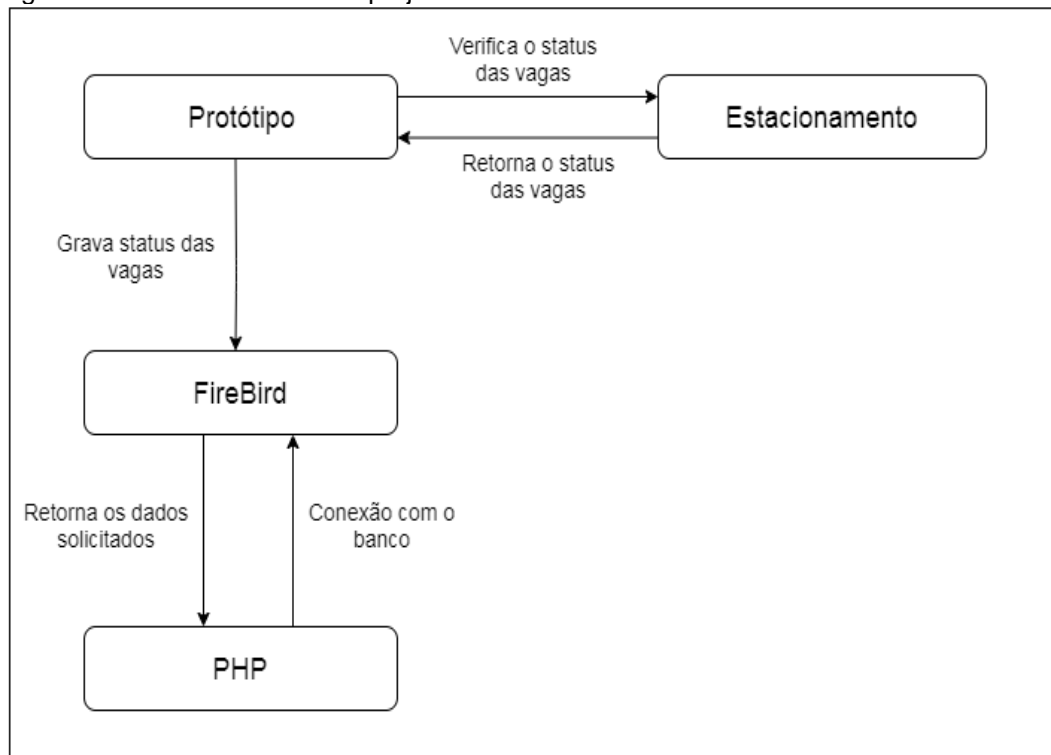
Descrição	Qtd	Função	Valor: US\$ Ref.: 03/2019
Câmera IP Bullet 720mp HD 2.8mm 20m Hikvision	1	Captura de imagens do estacionamento	245,15
Fonte chaveada POE 24v	1	Fonte de alimentação da câmera	89,78
<b>Total</b>			<b>334,93</b>

De acordo com os itens apresentados na tabela 5, torna-se possível a captura de imagens, para verificação do *status* de cada vaga de estacionamento proposta no projeto.

#### 4.1.2 Fluxo de dados e modelo de comunicação

Para o início do desenvolvimento, primeiramente estipulou-se um diagrama de fluxo de dados (figura 16), tendo como objetivo o mapeamento de todos os processos que o projeto é contido e todas as interconexões necessárias entre os mesmos. Com isso, facilitando uma visão global, facilitando o entendimento de todas as etapas a serem realizadas.

Figura 16 – Fluxo de dados do projeto



Fonte: Do autor.

O fluxo inicial estipulado, acontece no protótipo, onde este deve verificar o status de cada vaga do estacionamento e realizar o envio para o banco de dados. Sendo que, esta conexão é intermediada por uma DLL localizada no protótipo, tendo esta o objetivo de centralizar todas as operações que envolvem de alguma forma o uso da base de dados.

Por fim, a aplicação é a última parte do fluxo, ao qual nesta será possível realizar o gerenciamento das informações coletadas. Onde para a conexão com a base de dados, segue o mesmo formato citado anteriormente via arquivo PHP.

#### 4.1.3 Desenvolvimento do protótipo

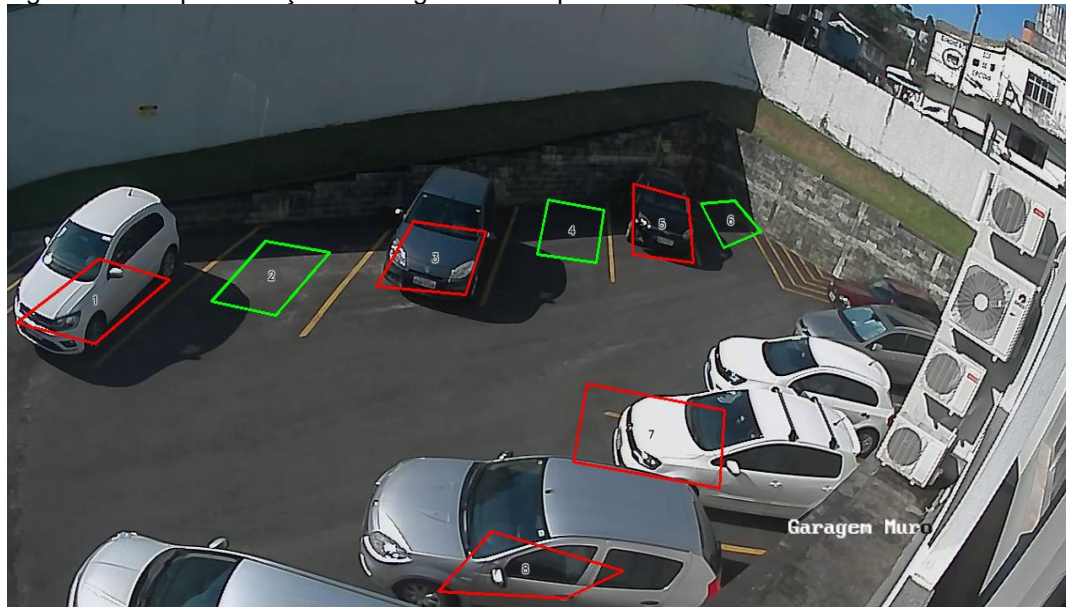
Para a montagem do projeto, foi preciso encontrar um editor de texto para uma melhor organização do código fonte, já que a linguagem de programação escolhida foi Python, a qual foi utilizada em conjunto com a biblioteca OpenCV. A linguagem em questão possui uma IDE, porém sua interface é simples e enxuta, sendo assim foi utilizado o editor de texto Atom, versão 1.40.1. Ferramenta esta, desenvolvida pelo GitHub sobre licença MIT (é uma licença de programas de computadores, criada pelo Instituto de Tecnologia de Massachusetts).

Após a escolha foi necessário baixar o editor de texto Atom, a linguagem Python e a biblioteca OpenCV para utilizá-las no projeto, realizada as configurações necessárias, ilustrado no apêndice 2, no sistema operacional Windows, fornecendo assim acesso aos funções disponibilizadas por ambas tecnologias.

Com a estrutura do projeto criada e pronta para o desenvolvimento, a etapa a seguir foi pesquisar e realizar testes para se familiarizar com a linguagem Python e a biblioteca OpenCV, desenvolvendo uma interface onde seria visualizado o vídeo capturado pela câmera em tempo real, em seguida o código para identificação das vagas. No decorrer desta etapa surgiu o maior desafio do projeto, o tratamento de imagens. A uma complexidade do OpenCV na teoria com na prática, mesmo possuindo diversas comunidades de discussão e uma vasta documentação, tornando mais difícil encontrar um caminho para iniciar o desenvolvimento do software cumprindo os objetivos do projeto.

Na figurar 17 observa-se o funcionamento do software, baseia-se nos pontos definidos para aplicação dos métodos descritos anteriormente, a função utilizada para realizar a interligação dos pontos das vagas é a *drawContours*. Também pode ser utilizada para desenhar qualquer forma, desde que tenha definido os pontos de limite, a aplicação deste método se baseia em três argumentos principais, o primeiro é a captura da imagem de origem, o segundo argumento é a leitura dos pontos definido em uma lista, o terceiro é o índice dos contornos importante quando se deseja desenhar contornos individuais e os demais argumentos se compõem da escala da cor desejada para a criação do retângulo, e também a espessura do mesmo.

Figura 17 – Representação das vagas com os pontos definidos



Fonte: Do autor.

Definido os pontos de cada vaga e os desenhado no vídeo, a etapa seguinte foi aplicar filtros a fim de poder identificar a alteração nas vagas demarcadas.

Dentre os métodos utilizados, o principal deles foi o método laplaciano, este filtro detecta bordas de objetos na imagem, que produz bordas finas da largura de um *pixel*, a imagem 18, ilustra a aplicação do método. Este método age em tempo real, juntamente com a captura do vídeo ao executar o software, em cada vaga são capturados os *frames* e aplicado a função *cv.Laplacian* que é executada a todo momento. Com isto, *frame a frame* é aplicado o filtro Laplace em tempo real na imagem capturada pela câmera.

Figura 18 – Detecção de bordas com o método Laplace



Fonte: Do autor.

Esta função é composta por métodos matemáticos capazes de identificar mudanças nos *pixels* da imagem. Para realizar a aplicação da função, são necessários dois argumentos, primeiro é a imagem original em *grayscale* e no segundo argumento é informado o tipo de dado da imagem final, neste projeto foi utilizado sessenta e quatro *bits*, para que todos os contornos da imagem sejam detectados. A ilustração 19 mostra a aplicação da função.

Figura 19 – Implementação do método Laplace

```

136     laplacian = cv2.Laplacian(roi_gray, cv2.CV_64F)
137     laplacian = np.uint8(np.absolute(laplacian))
138     points[:,0] = points[:,0] - rect[0]
139     points[:,1] = points[:,1] - rect[1]
140     delta = np.mean(np.abs(laplacian * parking_mask[ind]))

```

Fonte: Do autor.

Após a conclusão de toda a programação voltada para a identificação das vagas e alteração do *status* da mesma devido à ocupação dos automóveis, a próxima fase do projeto refere-se ao desenvolvimento da aplicação para a visualização dessas informações.

#### 4.1.4 Desenvolvimento da página WEB

Após o término de todo o desenvolvimento para identificação das vagas e alteração do status da mesma, deu-se início ao desenvolvimento da página web responsável por disponibilizar aos usuários a visualização das vagas livres e ocupadas.

Nas pesquisas realizadas sobre dispositivos móveis, percebeu-se que a utilização de *smartphones* é uma ferramenta muito útil e bastante utilizada no dia-a-dia das pessoas, e que todos possuem navegadores web, esse foi o principal motivo pela escolha para o desenvolvimento em uma página web, pois não se faz necessário reescrever todo o código fonte da mesma para diferentes sistemas operacionais que existem entre os *smartphones*.

A linguagem de programação escolhida nesta etapa foi a PHP com o objetivo para o desenvolvimento da parte lógica da página web. Para a criação de layout e demais funções, utilizado no projeto, foi utilizado o *framework* Angular, sendo este, uma plataforma utilizada para a construção de interfaces de aplicações desenvolvida pela Google, ao qual utiliza de tecnologias como *HTML*, *CSS* e *JavaScript*.

Com a linguagem definida, o próximo passo foi pensar de que maneira os usuários consigam visualizar as vagas e identifica-las com clareza, foi criado uma imagem da área do estacionamento escolhido, na mesma foi feita a marcação com números, como ilustra o apêndice C.

O apêndice A ilustra o layout da página web, onde cada vaga está representada em uma linha, as linhas possuem o número da vaga, status de livre ou ocupado e sua localidade caso esteja livre.

Finalizado o layout da aplicação web, o passo seguinte foi encontrar uma maneira para que houvesse a comunicação entre o software e a página web, obtendo as informações das vagas, e atualizando-as em tempo real.

#### 4.1.5 Integração entre software e aplicação web

A comunicação entre softwares e aplicativos pode-se realizar com a utilização de banco de dados ou *Web Services*, a escolha de qual tipo e ferramenta utilizar depende muito da situação e o objetivo de ambos. Neste projeto, decidiu-se utilizar o banco de dados *Firebird* para realizar a comunicação.

O *Firebird* é viável na utilização de grandes aplicações quanto para pequenas, nas aplicações de grande porte se faz necessário à aquisição de um equipamento exclusivo para o mesmo, pois necessita de processamento e armazenamento em escalas maiores, para pequenas aplicações que é o caso deste projeto não há necessidade de um equipamento exclusivo.

Este tipo de comunicação ocorre através de dll's do banco de dados instaladas no ambiente de desenvolvimento, feito à configuração no ambiente, basta realizar as funções de conexão no código fonte do software e na aplicação.

Na linguagem *Python*, são realizados os seguintes procedimentos de instalação, inicialmente executa-se a importação de *drivers* contendo todos os arquivos necessários para a comunicação, com o comando *import fdb*. Após a importação, é necessário abrir uma conexão com a base de dados através da função *fdb.connect*, nesta é informado os seguintes parâmetros: localização do banco de dados; usuário; senha. Estas duas últimas informações são definidas no momento em que o banco de dados é criado.

A figura 20 ilustra a utilização das funções de acesso ao banco de dados.

Figura 20 – Configuração de acesso ao banco de dados na linguagem Python

```
9 import fdb
10
11 conexao = fdb.connect(dsn='C:/TCC/PARK.fdb', user='sysdba', password='masterkey')
12 cursor = conexao.cursor()
```

Fonte: Do autor.

A configuração da comunicação do banco de dados na linguagem PHP possui algumas alternativas, dentre elas a função *ibase\_connect*, nesta é informado os seguintes parâmetros: localização do bando de dados; usuário; senha.

Em seguida, é utilizada a função *ibase\_query* responsável por realizar operações no banco de dados, como consultas, alterações exclusões de registros.

Nesta, são informados dados de acesso ao banco de dados, definidos na função anterior *ibase\_connect* e o comando SQL que se deseja aplicar.

Na figura 21 mostra à utilização das funções de acesso a base de dados.

Figura 21 – Configuração de acesso ao banco de dados na linguagem PHP

```

5   <?php
6
7       $conexao = ibase_connect("C:\TCC\PARK.fdb","SYSDBA","masterkey") or die
8           ('Erro : conexão banco');
9       $stmt = "select * from tb_estacionamento ";
10      $sth = ibase_query($conexao, $stmt);
11
12      ?>

```

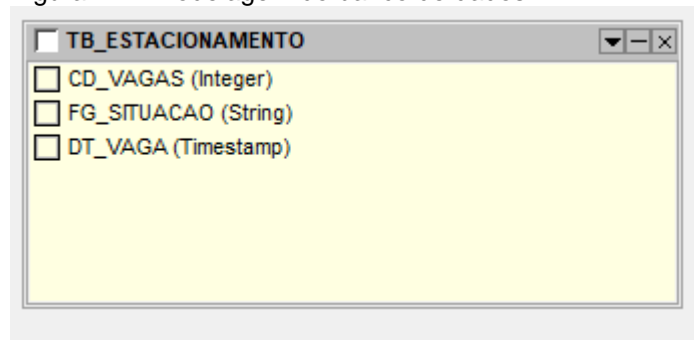
Fonte: Do autor.

Após a etapa de comunicação entre software e aplicação, iniciou-se a modelagem da tabela no banco de dados, ilustrada na figura 19, foi necessário à criação de uma tabela para o armazenamento das informações, pois se trata de um protótipo e somente da visualização de poucas informações.

Conforme ilustrada na figura 22, a modelagem é composta por uma tabela, que é:

- a) *tb\_estacionamento*: responsável por armazenar informações das vagas, como o código que armazena a *id* de cada vaga, *status* de livre ou ocupado e a data e hora em que foi alterado o status da vaga;

Figura 22 – Modelagem do banco de dados



Fonte: Do autor.

Com a modelagem finalizada, realizou-se a função de atualização do campo *status* na tabela, sendo este alterado para “L” se identificada à vaga como livre e “O” caso a vaga esteja ocupada, a figura 23 mostra a codificação da função *update* no corpo do software para a atualização dos campos da tabela mencionada.

Figura 23 – Comando SQL na linguagem Python

```

186 comando = "update tb_estacionamento set fg_situacao = '" + status + "
187         + ' where cd_vagas = " + str(park['id']) + " and fg_situacao <> '" + status + "'"
188 cursor.execute(comando)
189 conexao.commit()

```

Fonte: Do autor.

Na aplicação web, que fará o papel de cliente, configurou-se a função *select* como mostra a figura 24, este é responsável pela captura das informações incluídas na base de dados, possibilitando assim que o usuário tenha a visualização do *status* de cada vaga.

Figura 24 – Comando SQL na linguagem PHP

```

5  <?php
6
7      $conexao = ibase_connect("C:\Users\Rafael\Desktop\TCC\data\PARK.fdb", "SYSDBA", "masterkey") or die
8      ('Erro : conexão banco');
9      $stmt = "select * from tb_estacionamento ";
10     $sth = ibase_query($conexao, $stmt);
11
12     ?>

```

Fonte: Do autor.

#### 4.1.6 Funcionamento da aplicação web

Com o término da etapa de desenvolvimento da aplicação, a mesma ficou composta por apenas uma página, com uma funcionalidade específica. Está, possui a consulta dos dados referente às vagas e também a função de como chegar até elas.

No apêndice A observa-se as informações disponíveis aos usuários, as vaga são representadas por um código apresentado no início que cada linha, seguida do *status*, que de acordo com o seu valor altera a cor da linha para verde e apresentado o texto livre para as vagas disponíveis e também o ícone de localidade que mostra ao usuário qual a vaga o mesmo pretende se deslocar.

As linhas em vermelho representam as vagas ocupadas, descrito anteriormente todas as linhas iniciam com um código que representam o número de cada vaga, o *status* da vaga ocupada é representada por um ícone de atenção.

## 4.2 RESULTADOS E DISCUSSÕES

A proposta deste projeto de conclusão de curso teve como item de elaboração a aplicação do mesmo na área de estacionamento da UNESCO, porém foram encontrados alguns contratemplos para conclusão deste item. No dia 17 de outubro de 2019 a coordenação do curso de Ciência da Computação solicitou ao setor de obras da instituição a instalação da câmera de monitoramento no local desejado para a realização dos testes. Em contato com setor de obras sobre a instalação da mesma, obteve-se êxito da instalação no dia 31 de outubro de 2019 devido as fortes chuvas que ocorreram no mês outubro. Após a instalação, entrou-se em contato com o setor de Tecnologia da Informação nos dias 4 e 5 de novembro de 2019 solicitando o acesso à câmera para realização de testes, mas não se obteve sucesso devido a alguns empecilhos encontrados pelo setor em relação a segurança e outros aspectos. Fez-se necessário realizar os testes em outro local por definição própria, pois não havia possibilidade de aguardar devido ao prazo de entrega do projeto em questão.

Com a conclusão de todas as etapas necessárias para o projeto e os testes realizados nas funcionalidades disponibilizados por ele, tornou-se possível realizar uma análise relacionada aos resultados, com isso verificando a sua validade e aplicação.

Os testes realizados para obtenção de resultados foram realizados no estacionamento privado das Empresas Rio Deserto, sendo por esta liberada o acesso às imagens do estacionamento (apêndice D).

Os resultados preliminares que o algoritmo foi capaz de detectar a ausência e presença de veículos nas vagas de estacionamento, utilizando os conceitos de visão computacional, pré-processamento de imagens e a biblioteca OpenCV. Estes três conceitos unidos faz com que o resultado final seja mais preciso eliminando os pontos falsos positivos e falsos negativos.

Os testes foram realizados em oito vagas durante o período de 07 de novembro 2019 a 10 de novembro 2019, neste período foram realizados 246 testes, totalizando em 1.968 testes.

Para a medição da acurácia do software foram quantificados os seguintes itens: os verdadeiros positivos são as vagas que identificadas corretamente como ocupadas ou livres obtendo uma média de 34,67%, a taxa de falsos positivos são as

vagas que foram identificadas como ocupadas, mas não estavam com média de 4,13%, verdadeiros negativos são as vagas ocupadas, porém identificadas como livre média de 0,90% e os falsos negativos, é as vagas que não possuem carros e identificadas como livre com média de 60,28%, conforme descrito na tabela 3.

Tabela 2 – Tabela comparativa das taxas de erros e acertos

		Padrão Ouro (mundo real)	
		Carro presente	Carro não presente
Algoritmo	Carro presente	34,67%	4,13%
	Carro não presente	0,90%	60,28%

Fonte: Do Autor.

Com base nas informações ilustradas na tabela 2, fez-se possível o cálculo de algumas variáveis, como mostra a tabela 3. A primeira variável corresponde à sensibilidade da taxa de verdadeiros positivos, indica a capacidade de um teste detectar corretamente um determinado evento, que neste projeto são os eventos identificados corretamente como vagas ocupadas ou livres, obtendo-se uma média de 97,5%. Em seguida, mostra-se a especificidade, calcula-se a capacidade de um teste excluir corretamente um determinado evento, ou seja, as vagas ocupadas, porém identificadas como livre, média de 93,6%. E por último, mas não menos importante exiba-se a acurácia, tem como objetivo mostrar a precisão entre o valor obtido experimental e o valor verdadeiro, obtendo este uma média de 95%.

Tabela 3 – Resultados dos testes realizados

Variáveis	% (IC 95%)
Sensibilidade	97,5 (96,3 – 98,6)
Especificidade	93,6 (92,2 – 94,9)
Acurácia	95,0 (94,0 – 95,9)

Fonte: Do Autor.

Segundo Mexas (2014) em sua monografia de conclusão de curso, a qual o objetivo foi a realização de análise de algoritmos que possibilitam a identificação de objetos em imagens, para a modelagem de um algoritmo composto de identificação das áreas de estacionamento.

Para a medição da acurácia foi quantificado os seguintes itens: taxa de acerto sendo todos os pontos identificados corretamente dentro da área do estacionamento e os pontos falsos positivos, são estes os pontos que foram identificados fora da área de estacionamento e os falsos negativos definidos como os pontos que deveriam ser identificados, mas não foram (MEXAS, 2014).

Os testes foram realizados com 30 imagens, para a comparação dos resultados de processamento utilizou-se os seguintes modelagens: processamento de identificação do parque de estacionamento somente utilizando a modelagem de busca por carros estacionados, em seguida, a modelagem de busca por vagas sem veículos e por ultimo as duas modelagens em conjunto, obtendo os resultados ilustrados na figura 1 e 2 (MEXAS, 2014).

Figura 23 – Comparativo da taxa de acerto da identificação de pontos que correspondem à áreas de estacionamento

<b>Taxa de acertos dos pontos identificados</b>		
<b>Método</b>	<b>Média Aproveitamento</b>	
	<b>Taxa Acerto</b>	<b>Taxa FP</b>
<b>Busca por veículos estacionados dentro da área de asfalto</b>	98%	2%
<b>Busca por vagas sem veículos dentro da área de asfalto</b>	89%	11%
<b>Processo composto</b>	90%	10%

Fonte: Mexas (2014).

Figura 23 – Comparativo da taxa de acerto da identificação da área que correspondem à áreas de estacionamento

<b>Taxa de acertos das áreas identificados</b>			
<b>Método</b>	<b>Média de Aproveitamento</b>		
	<b>Taxa Acerto</b>	<b>Taxa FP</b>	<b>Taxa FN</b>
<b>Busca por veículos estacionados dentro da área de asfalto</b>	75%	5%	25%
<b>Busca por vagas sem veículos dentro da área de asfalto</b>	49%	1%	51%
<b>Processo composto</b>	85%	4%	15%

Fonte: Mexas (2014).

Com os resultados obtidos, Mexas (2014) relata que a identificação de pontos teve um maior acerto utilizando o método de busca para identificação de carros estacionados com taxa de acerto média de 98% e com falso positivo médio de 2%. Mas o objetivo do algoritmo é a identificação da área de todo o parque de estacionamento e o modelo composto pela identificação de carros estacionados e vagas sem veículos obteve um melhor aproveitamento com 85% de média de acerto e uma taxa de falso positivo de 15%.

Em comparação entre os dois projetos, onde Mexas (2014), obteve 85% de aproveitamento na identificação de carros estacionados e identificação de vagas sem veículos, sendo que o autor baseou-se na aplicação do software em imagens de alta resolução e o presente trabalho que consiste na aplicação de métodos para identificação de vagas ocupadas e livres em tempo real, obteve-se uma acurácia de 95%. Levando-se em conta que as variáveis encontradas em um ambiente de testes em tempo real são mais complexas e imprevisíveis, pode-se concluir que o presente trabalho obteve um bom resultado após as validações, sendo assim, pode ser aplicado em um ambiente real.

## 5 CONCLUSÃO

Com o término do trabalho, além de proporcionar aquisição de conhecimento relacionado a tecnologias utilizadas para a realização do protótipo, também forneceu noções básicas referentes aos conceitos relacionados à visão computacional e processamento de imagens, as quais podem ser aplicadas em diversas áreas, como em realidade virtual, identificação de objetos, reconhecimento facial, dentre outras. Estas tecnologias em conjunto com a criatividade possibilitam o desenvolvimento de inúmeras aplicações.

Com o conhecimento adquirido com a biblioteca OpenCV utilizada neste projeto para auxiliar no processamento de imagens, e com sua vasta comunidade, possibilitou o desenvolvimento do protótipo proposto.

Durante o desenvolvimento do projeto, algumas dificuldades foram encontradas com relação à parte prática, dentre elas podem ser citadas a complexidade na identificação de vagas e alteração do *status* da mesma, ao qual se fez necessário extensas pesquisas e tentativas até que se descobriu qual método a ser utilizado. O método de Laplace composto por métodos matemáticos pertence ao conjunto de algoritmos da biblioteca OpenCV, tem como objetivo em sua aplicação determinar os contornos dos objetos presentes na imagem.

Relativo ao conhecimento adquirido e etapas concluídas, foi possível o desenvolvimento de um protótipo para a detecção de vagas de estacionamento através de imagens capturadas por uma câmera IP, aplicando o processamento de imagens com auxílio da biblioteca OpenCV, verificando o *status* se estão livres ou ocupadas, e também uma aplicação WEB para visualização das vagas.

Com base nos testes realizados, obteve-se um resultado satisfatório quanto à integração entre todas as tecnologias implementadas e em comparação com o projeto de Antonio Henrique Mexas onde seu objetivo foi à identificação de carros e vagas de estacionamento através de imagens estáticas de alta resolução, obtendo uma taxa de acertos de 98% na busca por veículos nas vagas de estacionamento e 89% por vagas sem veículos, levando-se em conta que as variáveis encontradas em um ambiente de testes em tempo real são mais complexas e imprevisíveis, obteve-se uma acurácia de 95%, pode-se concluir que o presente trabalho obteve um bom resultado após as validações, sendo assim, pode ser aplicado em um ambiente real.

De acordo com o conhecimento obtido e nos resultados adquiridos com o projeto, sugere-se como trabalhos futuros:

- a) Aplicação de inteligência artificial para identificação de vagas e reconhecimento de veículos nas mesmas;
- b) Aplicar a api Google Maps na aplicação WEB para fornecer uma melhor identificação da vaga e deslocamento até a mesma;
- c) Desenvolvimento de uma aplicação mobile.

## REFERÊNCIAS

ALHAK, Said Hikmat Abd. **Estacionameto Inteligente**. 2011. 76 p. TCC(Graduação) - Curso de Engenharia da Computação, Centro Universitário de Brasília – Uniceub Fatecs – Faculdade de Tecnologia e Ciências Sociais Aplicadas Curso de Engenharia de Computação, Brasília, 2011. Disponível em: <<https://repositorio.uniceub.br/jspui/bitstream/123456789/3139/2/20516680.pdf>> Acesso em: 24 nov. 2018.

ALMEIDA, Claudio de B. **Câmeras IP - 1ª parte**. Instituto CFTV. 2015. Disponível em: <<http://www.institutocftv.com.br/cameras-ip.html>> Acesso em: 05 fev. 2019.

BORGES, Eder. **A mobilidade urbana centrada no uso do automóvel**. 2012. Disponível em: <<http://periodicos.uem.br/ojs/index.php/Percurso/article/view/24798>>. Acesso em: 01 jun. 2019.

BARWINSKI, Luísa. **A World Wide Web completa 20 anos, conheça como ela surgiu**. 2009. Disponível em: <<https://www.tecmundo.com.br/historia/1778-a-world-wide-web-completa-20-anos-conheca-como-ela-surgiu.htm>> Acesso em: 05 out. 2018.

BRADSKI, Gary; KAEHLER, Adrian. **Learning OpenCv**. Sebastopol: O'reilly Media, 2008. 571 f. Disponível em: <<https://www.bogotobogo.com/cplusplus/files/OReilly%20Learning%20OpenCV.pdf>>. Acesso em: 29 jan. 2019.

BRAHMBHATT, Samarth. **Practical OpenCv**. New York: Technology In Action, 2013. 229 p. Disponível em: <<https://www.amazon.com/Practical-OpenCV-Technology-Samarth-Brahmbhatt/dp/1430260793>>. Acesso em: 29 set 2019.

BRUNELLI, Roberto. **Template Matching Techniques in Computer Vision: Theory and Practice**. 2000. Disponível em: <[https://www.researchgate.net/publication/272496016\\_Template\\_Matching\\_Techniques\\_in\\_Computer\\_Vision\\_Theory\\_and\\_Practice](https://www.researchgate.net/publication/272496016_Template_Matching_Techniques_in_Computer_Vision_Theory_and_Practice)>. Acesso em: 12 out 2019.

CARVALHO, Carlos Henrique Ribeiro de. **Mobilidade Urbana: Avanços, desafios e perspectivas**. Editora Multimidia. 2010. 20 p. Disponível em: <[http://www.ipea.gov.br/portal/images/stories/PDFs/livros/livros/160920\\_estatuto\\_cidade\\_cap14.pdf](http://www.ipea.gov.br/portal/images/stories/PDFs/livros/livros/160920_estatuto_cidade_cap14.pdf)>. Acesso em: 29 maio 2019.

CHAUDHARI, Aniruddha. **Advantages and Disadvantages of YAML over XML and JSON**. 2017. Disponível em: <<https://www.csestack.org/advantages-disadvantages-yaml/>>. Acesso em: 05 jun. 2018.

CORCOBADO, Ángel Miguel. **Trânsito de São Paulo está na lista dos cinco maiores congestionamentos da história**. 2017. Disponível em: <[https://brasil.elpais.com/brasil/2017/05/08/internacional/1494262753\\_775936.html](https://brasil.elpais.com/brasil/2017/05/08/internacional/1494262753_775936.html)>. Acesso em: 16 set. 2018.

CUNHA, André Luiz Barbosa Nunes da. **Sistema Automático para Obtenção de Parâmetros do Tráfego Veicular a partir de Imagens de Vídeo usando OpenCv**. 2013. 128 p. Curso de Escola de Engenharia de São Carlos, Universidade de São Paulo, São Paulo, 2013. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/18/18144/tde-19112013-165611/pt-br.php>>. Acesso em: 18 maio 2019.

FONSECA, Rúben; SIMÕES, Alberto. **Alternativas ao XML: YAML e JSON**. Departamento de Informática, Universidade do Minho. 2007. Disponível em: <<http://xata.fe.up.pt/2007/papers/3.pdf>>. Acesso em: 30 maio 2019.

GUGELMIN, Felipe. 2014. **Internet: qual a diferença entre os protocolos UDP e TCP?**. Disponível em: <https://www.tecmundo.com.br/internet/57947-internet-diferenca-entre-protocolos-udp-tcp.htm>. Acesso em: 23 ago. 2018.

GONZALEZ, Rafael C.; WOODS, Richard E. **Processamento De Imagens Digitais**, São Paulo: Edgard Blücher, 2000. Disponível em: <<https://edoc.pub/processamento-digital-de-imagens-3ed-pdf-free.html>>. Acesso em: 23 ago. 2018.

GRANGE, Sébastien; FONG, Terrence; BAUR, Charles. **TLIB: a Real-time Computer Vision Library for HCI**. Sydney: Swiss Federal Institute Of Technology, 2003. Disponível em: <[https://www.ri.cmu.edu/pub\\_files/pub4/grange\\_sebastien\\_2003\\_1/grange\\_sebastien\\_2003\\_1.pdf](https://www.ri.cmu.edu/pub_files/pub4/grange_sebastien_2003_1/grange_sebastien_2003_1.pdf)>. Acesso em: 02 nov. 2018.

GREATEK, Blog. **Distância do Infravermelho**. 2017. Disponível em: <<http://blog.greatek.com.br/2017/12/19/qual-a-funcao-do-infravermelho-na-camera-de-seguranca-no-cftv/>>. Acesso em:

GUALBERTO, Daniel Rocha. **Em rumo a um sistema automático de controle de acesso de veículos automotivos: reconhecimento de caracteres em placas de veículos**. 2010. 64 p. TCC (Graduação) - Curso de Ciência da Computação, Universidade Federal de Ouro Preto, Ouro Preto, 2010. Disponível em: <<http://www.decom.ufop.br/menotti/monoll102/files/BCC391-102-aom-07.1.4176-DanielRochaGualberto.pdf>>. 09 set. 2018.

G1. **Venda de veículos sobe 26,6% em fevereiro, diz Fenabreve**. Disponível em: <<https://g1.globo.com/carros/noticia/2019/03/01/venda-de-veiculos-sobe-266-em-fevereiro-diz-fenabreve.ghtml>>. Acesso em: 15 set. 2018.

HAMILTON, Emily. **Como o estacionamento grátis piora o congestionamento**. 2017. Disponível em: <<https://caosplanejado.com/como-o-estacionamento-gratis-piora-o-congestionamento/>>. Acesso em: 10 set. 2018.

JOHNSON, Thienne M. **Java para Dispositivos Móveis: Desenvolvendo Aplicações com J2ME**. São Paulo: Novatec, 2007. 336 p. Disponível em: <<https://docplayer.com.br/7128656-Java-para-dispositivos-moveis-thienne-m-johnson-novatec-desenvolvendo-aplicacoes-com-j2me.html/>>. Acesso em: 02 fev. 2019.

LUIZ, Ricardo. **Conheça os diferentes tipos de Resoluções em CFTV e qual a diferença de 1080N para 1080P**. 2016. Disponível em: <<http://ricardoluiz.com.br/resolucao/>>. Acesso em: 29 jan. 2019.

MACORATTI, José Carlos. **XML – Introdução e conceitos básicos**. 2010. Disponível em: <<http://www.macoratti.net/xml.htm>>. Acesso em: 08 maio 2019.

MARQUES FILHO, Ogê; VIEIRA NETO, Hugo. **Processamento Digital de Imagens**. Rio de Janeiro: Brasport, 1999. 410 p. Disponível em: <<http://www.ogemarques.com/wp-content/uploads/2014/11/pdi99.pdf>>. Acesso em: 07 maio 2019.

MAZUCHETTI, Muriel Rampinelli. **ESTUDO DE FRAMEWORK MULTIPLATAFORMA APLICADO AO DESENVOLVIMENTO DE UM PROTÓTIPO DE APLICATIVO MOBILE HÍBRIDO PARA O CONTROLE FINANCEIRO PESSOAL**. 2015. 79 f. TCC (Graduação) - Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, 2015.

MEDEIROS, Henrique. **92% dos brasileiros possuem ou usam smartphones com frequência**. 2018. Disponível em: <<https://www.mobiletime.com.br/noticias/18/10/2018/92-dos-brasileiros-possuem-ou-usam-smartphones-com-frequencia/>>. Acesso em: 18 maio 2019.

MELO, João Eduardo. **Estacionamento privado ou público como diferenciar**. 2015. Disponível em: <<https://www.rotaryclubdistrito4500.com.br/joomla30/index.php/462-Estacionamento-privado-ou-p%C3%BAblico-como-diferenciar/>>. Acesso em: 01 jun. 2018.

MEXAS, Antonio Henrique. **PROCESSO DE RECONHECIMENTO NÃO SUPERVISIONADO DE ÁREAS DE ESTACIONAMENTO**. 2014. 71 p. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Universidade Presbiteriana Mackenzie, São Paulo, 2014. Disponível em: <<http://tede.mackenzie.br/jspui/bitstream/tede/1441/1/Antonio%20Henrique%20Mexas.pdf>>. Acesso em: 15 nov. 2018.

MILLER, Erick G. Learned. **Introduction to Computer Vision**. Amherst, 2011. 11 p. Disponível em: <[https://people.cs.umass.edu/~elm/Teaching/Docs/IntroCV\\_1\\_19\\_11.pdf](https://people.cs.umass.edu/~elm/Teaching/Docs/IntroCV_1_19_11.pdf)>. Acesso em: 11 nov. 2018.

MONTEIRO, Geovana; VACCARI, Rafaella. **A tecnologia associada ao nosso dia a dia**. 2016. Disponível em: <<http://portal.metodista.br/rpcom/mundo-corporativo/a-tecnologia-associada-ao-nosso-dia-a-dia>>. Acesso em: 01 jun. 2018.

NESBITT, Scott. **Browsing the web with Min, a minimalist open source web browser**. 2018. Disponível em: <<https://opensource.com/article/18/10/min-web-browser>>. Acesso em: 13 nov. 2018.

NETO, Gerlane. **Mobilidade Urbana**. 2018. Disponível em:

<<https://www.polemicaparaiba.com.br/paraiba/mobilidade-urbana-ciclovia-e-uma-opcao-de-locomocao-em-cidades-com-muito-congestionamento>>. Acesso em: 29 jan. 2019.

OLIVEIRA, Ricardo. **A internet e os navegadores**. 2014. Disponível em: <<https://administradores.com.br/artigos/a-internet-e-os-navegadores>> Acesso em: 15 out. 2018.

ORGERA, Scott. **What Is a Web Browser?**. 2019. Disponível em: <https://www.lifewire.com/what-is-a-browser-446234>> Acesso em: 04 nov. 2018.  
PEDOT, Lauro César. **Parada e estacionamento de veículo: qual a diferença?**. 2010. Disponível em: < <http://abordagempolicial.com/2010/04/parada-e-estacionamento-de-veiculo-qual-a-diferenca/>> Acesso em: 11 nov. 2018.

PERCÍLIA, Eliene. **Comunicação de Dados**. Brasil Escola. Disponível em: <https://brasilecola.uol.com.br/informatica/comunicacao-dados.htm>. Acesso em: 31 out. 2018.

PILLOU, Jean François. **Protocolo TCP**. CCM. 2017. Disponível em: <<https://br.ccm.net/contents/284-o-protocolo-tcp>> Acesso em: 26 out. 2018.

PRADO, Marcos Gomes. **Planejamento de trajetória para estacionamento de veículos autônomos**. 2013. 84 p. Dissertação (Mestrado) - Curso de Ciências da Computação e Matemática Computacional, Icmc-usp, São Carlos, 2013. Disponível em: < <http://www.teses.usp.br/teses/disponiveis/55/55134/tde-13052013-101339/publico/marcospradorevisada.pdf> > Acesso em: 10 set. 2018.

RIBEIRO, Breno. **Entenda a evolução dos smartphones e como a sociedade se comporta com novas tecnologias**. 2018. Disponível em: <<https://www.folhavoria.com.br/geral/noticia/05/2018/entenda-a-evolucao-dos-smartphones-e-como-a-sociedade-se-comporta-com-novas-tecnologias>>. Acesso em: 15 nov. 2018.

RIOS, Luiz Romário Santana. **Visão Computacional**. Bahia. 8 p. Disponível em: <[http://homes.dcc.ufba.br/~luizromario/Apresenta%C3%A7%C3%A3o%20de%20IA/Artigo%20\(final\).pdf](http://homes.dcc.ufba.br/~luizromario/Apresenta%C3%A7%C3%A3o%20de%20IA/Artigo%20(final).pdf)>. Acesso em: 05 out. 2018.

RODRIGUE, Jean Paul. **Urban Mobility and its Evolution**. 2017. Disponível em: <[https://transportgeography.org/?page\\_id=4617](https://transportgeography.org/?page_id=4617)>. Acesso em: 18 out. 2018.

ROSEBROCK, Adrian. **OpenCV center of contour**. 2016. Disponível em: < <https://www.pyimagesearch.com/2016/02/01/opencv-center-of-contour/>>. 06 out. 2018.

ROSEBROCK, Adrian. **Blur detection with OpenCV**. 2016. Disponível em: < <https://www.pyimagesearch.com/2015/09/07/blur-detection-with-opencv/>>. Acesso em: 16 nov. 2018.

- SAMBUGARO, Adriano. **Tudo o que você precisa saber sobre CFTV e câmeras de segurança**. 2016. Disponível em: <<http://blog.gunnebo.com.br/tudo-o-que-voce-precisa-saber-sobre-cftv-e-cameras-de-seguranca>>. Acesso em: 05 out. 2018.
- SANTA CATARINA. DETRAN-SC. **Frota de Veículos no Estado**. Disponível em: <<http://consultas.detrannet.sc.gov.br/Estatistica/Veiculos/>>. Acesso em: 25 out 2018.
- SELVENCA, Juliano Neres; COUTO, Tiago Silva. **Sistema de reconhecimento de vagas em estacionamento por meio de câmeras**. 2012. Disponível em: <<https://www.escavador.com/patentes/78574/sistema-de-reconhecimento-de-vagas-em-estacionamento-por-meio-de-cmeras?page=1>>. Acesso em: 10 nov. 2018.
- SEN, Saptarshi; AHMED, Mokaddes Ali; DAS Debasish. **A Case Study on On-Street Parking Demand Estimation for 4-Wheelers in Urban CBD**. 2016. 5 p. Disponível em: <[https://www.researchgate.net/publication/307584016\\_A\\_Case\\_Study\\_on\\_On-Street\\_Parking\\_Demand\\_Estimation\\_for\\_4-Wheelers\\_in\\_Urban\\_CBD](https://www.researchgate.net/publication/307584016_A_Case_Study_on_On-Street_Parking_Demand_Estimation_for_4-Wheelers_in_Urban_CBD)>. Acesso em:
- SINGH, Atul Krishna; KANG, Pankaj. **Laplacian of Gaussian (LoG)**. 2019. Disponível em: <<https://theailearner.com/tag/cv2-laplacian/>>. Acesso em: 09 mar. 2019.
- STALLINGS, William; BROWN, Lawrie. **Segurança de Computadores: Princípios e Práticas**. 2. ed. São Paulo: Elsevier, 2013. ISBN 9788535264494.
- STOREY, Jonathan. **The world's new vehicle market: Outlook for 2018**. Washington Buildings, Martin Kahl, 2017. 19 p. Disponível em: <<http://www.acs-giz.si/resources/files/research-the-worlds-new-vehicle-market-outlook-for-2018.pdf>>. Acesso em: 16 nov. 2018.
- SOARES, Ana. **Introdução à imagem digital, definição de pixel e cores digitais**. 2010. Disponível em: <<https://anasoares1.wordpress.com/2010/11/22/introducao-a-imagem-digital-definicao-de-pixel-e-cores-digitais/>>. Acesso em: 29 abr. 2019.
- SOUSA, Kelly Aparecida Oliveira. **USO DE VISÃO COMPUTACIONAL EM DISPOSITIVOS MÓVEIS PARA AUXÍLIO À TRAVESSIA DE PEDESTRES COM DEFICIÊNCIA VISUAL**. 2013. 85 p. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Universidade Presbiteriana Mackenzie, São Paulo, 2013. Disponível em: <[http://iris.sel.eesc.usp.br/wvc/Anais\\_WVC2012/pdf/97956.pdf](http://iris.sel.eesc.usp.br/wvc/Anais_WVC2012/pdf/97956.pdf)> 20 nov. 2018.
- SZELISKI, Richard. **Computer Vision: Algorithms and Applications**. 2010. 979 p. Disponível em: <[http://szeliski.org/Book/drafts/SzeliskiBook\\_20100903\\_draft.pdf](http://szeliski.org/Book/drafts/SzeliskiBook_20100903_draft.pdf)>. Acesso em: 04 nov. 2018.
- TELECO. **Estatísticas de Celulares no Brasil**. 2018. Disponível em: <http://www.teleco.com.br/ncel.asp>. Acesso em: 01 out. 2018.

ZANETTE, Jackson Guizzo. **RECONHECIMENTO AUTOMÁTICO DE PLACAS VEICULARES UTILIZANDO A TECNOLOGIA OCR E A PLATAFORMA RASPBERRY PI APLICADA NA FISCALIZAÇÃO ELETRÔNICA DE RODOVIAS.** 2016. 117 p. TCC (Graduação) - Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, 2016.

ZHOU, Huiyu; WU, Jiahua; ZHANG, Jianguo. **Digital Image Processing.** [S.l]: Book Boon, 2010. 71 p. Disponível em:  
<<https://pdfs.semanticscholar.org/17e5/d11c00370444d84ef351515e536df0f9abf8.pdf>>. Acesso em: 06 out. 2018.

WILKEN, Jeremy. **Ionic in Action: HYBRID MOBILE APPS WITH IONIC AND ANGULARJS.** Shelter Island: Manning, 2016. 282 p. Disponível em:  
<<http://pepa.holla.cz/wp-content/uploads/2016/12/Ionic-in-Action.pdf>>. Acesso em: 10 nov. 2018.

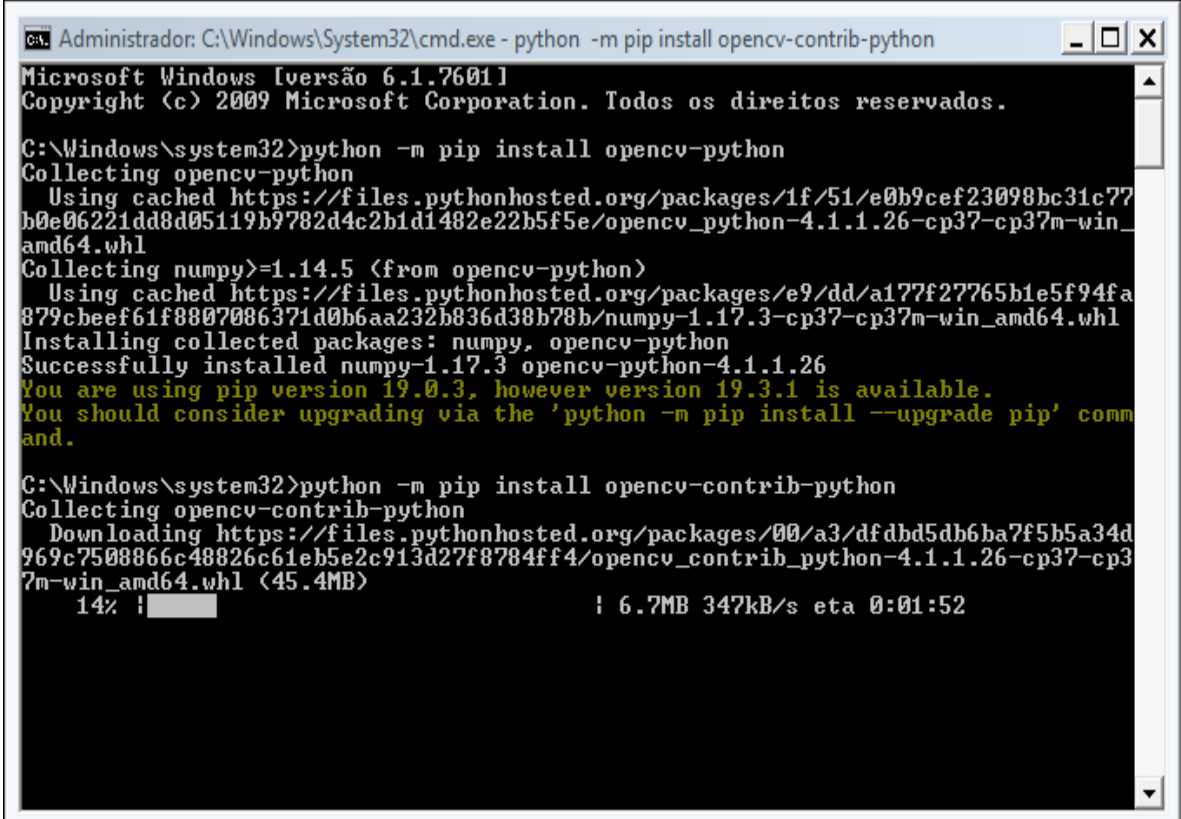
**APÊNDICE(S)**

## APÊNDICE A – LAYOUT DA APLICAÇÃO WEB

TCC - Rafael Michels Américo

1	⚠	
2	Livre	📍
3	⚠	
4	⚠	
5	Livre	📍
6	⚠	
7	Livre	📍
8	⚠	

## APÊNDICE B – CONFIGURAÇÃO DA BIBLIOTECA OPENCV



```
Administrador: C:\Windows\System32\cmd.exe - python -m pip install opencv-contrib-python
Microsoft Windows [versão 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.

C:\Windows\system32>python -m pip install opencv-python
Collecting opencv-python
  Using cached https://files.pythonhosted.org/packages/1f/51/e0b9cef23098bc31c77b0e06221dd8d05119b9782d4c2b1d1482e22b5f5e/opencv_python-4.1.1.26-cp37-cp37m-win_
amd64.whl
Collecting numpy>=1.14.5 (from opencv-python)
  Using cached https://files.pythonhosted.org/packages/e9/dd/a177f27765b1e5f94fa879cbeef61f8807086371d0b6aa232b836d38b78b/numpy-1.17.3-cp37-cp37m-win_
amd64.whl
Installing collected packages: numpy, opencv-python
Successfully installed numpy-1.17.3 opencv-python-4.1.1.26
You are using pip version 19.0.3, however version 19.3.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' comm
and.

C:\Windows\system32>python -m pip install opencv-contrib-python
Collecting opencv-contrib-python
  Downloading https://files.pythonhosted.org/packages/00/a3/dfdbd5db6ba7f5b5a34d969c7508866c48826c61eb5e2c913d27f8784ff4/opencv_contrib_python-4.1.1.26-cp37-cp3
7m-win_
amd64.whl (45.4MB)
    14% |██████████          | 6.7MB 347kB/s eta 0:01:52
```

## APÊNDICE C - ARTIGO CIENTÍFICO

**Controle de vagas de estacionamento a partir do tratamento de imagens em tempo real com a biblioteca OpenCV****Rafael Michels Américo<sup>1</sup>, Sérgio Coral<sup>2</sup>**

<sup>1</sup>Acadêmico do Curso de Ciência da Computação - Unidade Acadêmica de Ciências, Engenharias e Tecnologias - Universidade do Extremo Sul Catarinense (UNESC) - Criciúma – SC

<sup>2</sup>Professor do Curso de Ciência da Computação - Unidade Acadêmica de Ciências, Engenharias e Tecnologias - Universidade do Extremo Sul Catarinense (UNESC) - Criciúma - SC

rafael.m.americo@gmail.com, sergio@unesc.net

**Abstract.** *The disorderly growth of vehicles on the street is a major problem in all countries, and has been increasing significantly over the years, generating major traffic congestion and also affecting the search for parking spaces. With the evolution of technology, solutions have emerged to alleviate the inconvenience by looking for a parking space, but there are not a lot of options, and have a high cost of deployment. Given this context, the present work aims to research and develop a prototype capable of identifying parking spaces using computer vision concepts, applying image processing with the help of the OpenCV library; this library is specific to this type of application branch in open source licensed image processing. The software developed in Python language captures the parking lot image in real time through an IP camera and it applies image processing with the help of filters and methods of the library. It is possible to define the parking spaces through image points and these are interpreted by the source code and then present them. The identification of cars to change the status of vacancies was obtained through the Laplace method that is also part of the library. The frequent use of smartphone in everyday life has become an indispensable tool due to the resources it offers, it is also proposed to develop a WEB page so that the user can view available and busy vacancies.*

**Resumo.** *O crescimento desordenado de veículos nas ruas é um grande problema em todos os países, e vem aumentando significativamente com o passar dos anos, gerando grandes congestionamentos em vias e afetando também na procura por locais para estacionar o veículo. Com a evolução da tecnologia surgiram soluções para amenizar o transtorno pela procura de uma vaga de estacionamento, porém não são muitas opções, e possuem um alto custo de implantação. Diante deste contexto, o presente trabalho visa a pesquisa e o desenvolvimento de um protótipo capaz de identificar as vagas de estacionamento utilizando conceitos de visão computacional, aplicando o tratamento de imagens com auxílio da biblioteca OpenCV; biblioteca está específica para este tipo de ramo de aplicação em processamento de imagens com licença de código aberto. O software desenvolvido na linguagem Python captura a imagem do estacionamento em tempo real através de uma câmera IP e este aplica um processamento de imagens com auxílio de filtros*

*e métodos que compõem a biblioteca, sendo possível definir as vagas de estacionamento através de pontos da imagem e estes interpretados pelo código fonte para então apresenta-los. A identificação dos carros para alteração do status das vagas foi obtido através do método de Laplace que também faz parte da biblioteca. A utilização frequente de smartphone no dia-a-dia tornou-se uma ferramenta indispensável devido aos recursos que ele oferece, também é proposto desenvolvimento de uma página WEB para que o usuário possa visualizar as vagas disponíveis e ocupadas.*

## **1. Introdução**

O avanço da tecnologia proporcionou velocidade e dinamismo em tarefas realizadas no dia-a-dia para as pessoas. Por exemplo, ao pegar um ônibus ou outro meio de locomoção para um determinado destino pode haver o envolvimento da tecnologia, assim, pode se dizer que a influência desta no ritmo de vida da sociedade atual é enorme (MONTEIRO; VACCARI, 2016).

Com o aumento da necessidade da locomoção mais ágil e eficiente das pessoas para locais de trabalho, moradia, lazer, entre outros locais que fazem parte da rotina diária, proporcionou um grande aumento dos veículos individuais e particulares em circulação nas cidades. Em algumas situações de locomoção se é necessário que o condutor pegue as vias mais cedo, horas antes do planejado. Está necessidade se da por conta dos grandes engarrafamentos e a dificuldade de encontrar uma vaga de estacionamento (NETO, 2018).

A procura por vaga de estacionamento nos grandes centros é diária, muitos condutores optam por utilizar um serviço de estacionamento privado, onde deixam seu carro enquanto realizam suas tarefas diárias, porém, ainda assim encontram estacionamentos lotados, devido ao grande fluxo de carros nos estacionamentos (HAMILTON, 2017).

Segundo Medeiros (2018) 92% dos brasileiros possuem ou usam um smartphone. Os smartphones se tornaram uma ferramenta essencial para as pessoas, além de possuírem grande poder de processamento integram vários recursos. Estes dispositivos possibilitam que as pessoas realizem ligações, conexão com a internet com alta disponibilidade e a utilização de aplicativos que proporcionam agilidade em suas tarefas diárias, tais como GPS, aplicativos bancários, entre outras funções que o dispositivo pode proporcionar (RIBEIRO, 2018).

O desenvolvimento de aplicativos na área de visão computacional conta na maioria das vezes com algum tipo de biblioteca, tendo objetivo de auxiliar como um componente chave baseado na visão de aplicações. Existem inúmeras bibliotecas, cada uma delas com suas características, muitas delas são complicadas ou limitadas à compatibilidade de hardware/software, estes fatores tendem a impedir o desenvolvimento de aplicações de pesquisa (GRANGE; FONG; BAUR, 2003, tradução nossa).

A biblioteca OpenCV, aplicada para o desenvolvimento do protótipo no presente trabalho, é um componente multiplataforma, totalmente livre sem fins lucrativos, podendo ser aplicada para fins acadêmicos quanto comercialmente ou para qualquer pessoa tenha o interesse em utilizá-la (BRADSKI; KAEHLER, 2008, tradução nossa).

A Visão Computacional é uma área da computação que estuda métodos para a extração de informações dos objetos em uma imagem, como velocidade, formas, entre outras. Todas as extrações são feitas para alcançar um determinado objetivo. Diferente do processamento de imagens, esta área consiste em transformar uma imagem em outra imagem,

através da manipulação e obtenção dos dados de uma imagem e do uso destas para diferentes objetivos (RIOS, 2010).

## 2. Mobilidade urbana

A mobilidade urbana é um tema importante quando se discute sobre o desenvolvimento urbano e qualidade de vida da população. Há diversos conceitos que a definem, dentre eles podemos citar o deslocamento das pessoas e das mercadorias nos centros urbanos, sendo este realizado por meio de transportes públicos ou privados (RODRIGUES, 2017, tradução nossa).

No século XX, o automóvel se tornou uma figura importante na cultura, da sociedade e da economia mundial. Mas com o passar do tempo o número de automóveis teve um crescimento de forma significativa, e, sem um aumento proporcional no número das estradas, surgiram os grandes engarrafamentos no século seguinte, e assim aumentando também a poluição nas cidades (BORGES, 2012).

## 3. Processamento de imagens

O Processamento de Imagens (PID) é a tecnologia que realiza o tratamento de imagens através de algoritmos computacionais, podendo ser encontrada em sistemas inteligentes ou robóticos, sistemas de detecção de objetos, entre outros. Uma imagem digital é composta por vários *pixels*, sendo milhares de pontos que combinam as cores, vermelho, verde e azul para formar as cores do mundo real. O PID tem como seu principal objetivo auxiliar humanos na obtenção de imagens de alta qualidade ou características da imagem original (ZHOU; WU; ZHANG, 2010, tradução nossa).

Com o crescimento da área de PID tornou-se viável o desenvolvimento de aplicações de análise automática por computador de informações de uma cena, podendo ser definida como “análise de imagens”, “reconhecimento de padrões” ou “visão computacional”. Um sistema de processamento digital de imagem é composto por alguns processos que se aplicam em sistemas de baixo custo até estações de trabalho sofisticadas. Podemos citar estes processos como: aquisição, armazenamento, processamento e exibição. A figura 1 ilustra a sequência destes processos (GONZALEZ; WOODS, 2000).

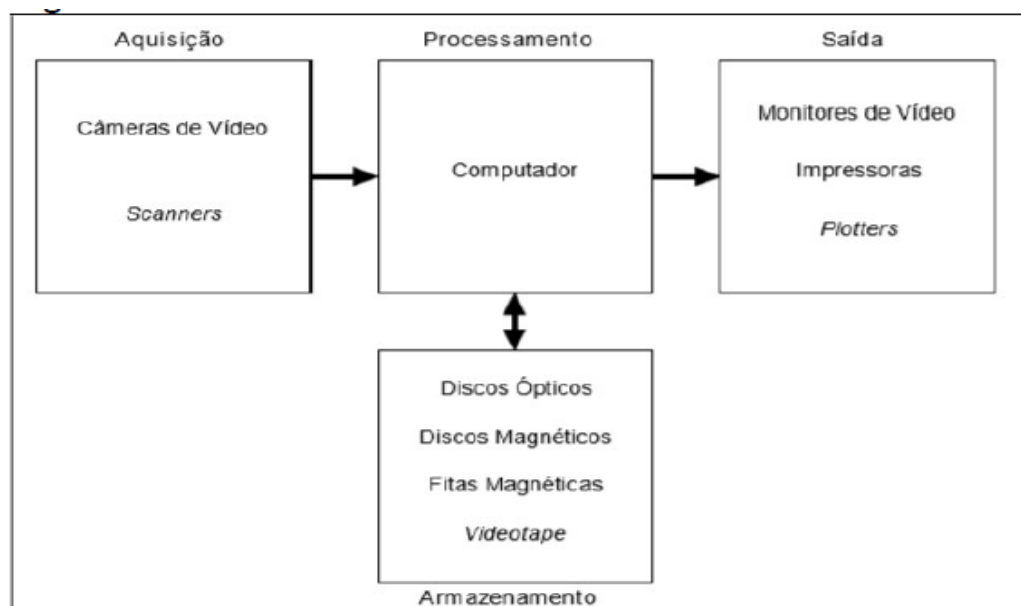


Figura 1. Fluxo de um sistema PID

### 3.1 Visão Computacional

Visão computacional pode ser observada como a ciência que pesquisa tecnologias que concede as máquinas à capacidade de extrair e processar informações através de imagens (BRUNELLI, 2009, tradução nossa). Considerada como um amplo e complexo campo de estudo, a inteligência computacional, tem sido uma área de pesquisa bastante ativa e crescente nas últimas décadas (MILLER 2011, tradução nossa).

No ponto de vista da engenharia é o processo de automatizar as tarefas que o sistema visual humano desempenha, assim como um médico pode identificar um tumor em uma tomografia ou uma pessoa reconhecer o próprio rosto em uma fotografia (SZELISKI, 2010, tradução nossa).

A primeira etapa executada pela visão computacional se inicia na aquisição, ilustrada na figura 2, sendo está responsável em converter a imagem em uma representação numérica em que o computador possa interpretar. Esta aquisição é composta por dois elementos, o primeiro é um dispositivo sensível a uma faixa de energia no espectro eletromagnético (visível ou invisível aos humanos) que produz um sinal elétrico analógico e o segundo um digitalizador que converte este sinal em uma informação binária. Este processo é executado através de uma câmera, ou outro equipamento eletrônico capaz de capturar imagens (MARQUES FILHO; VIEIRA NETO, 1999).

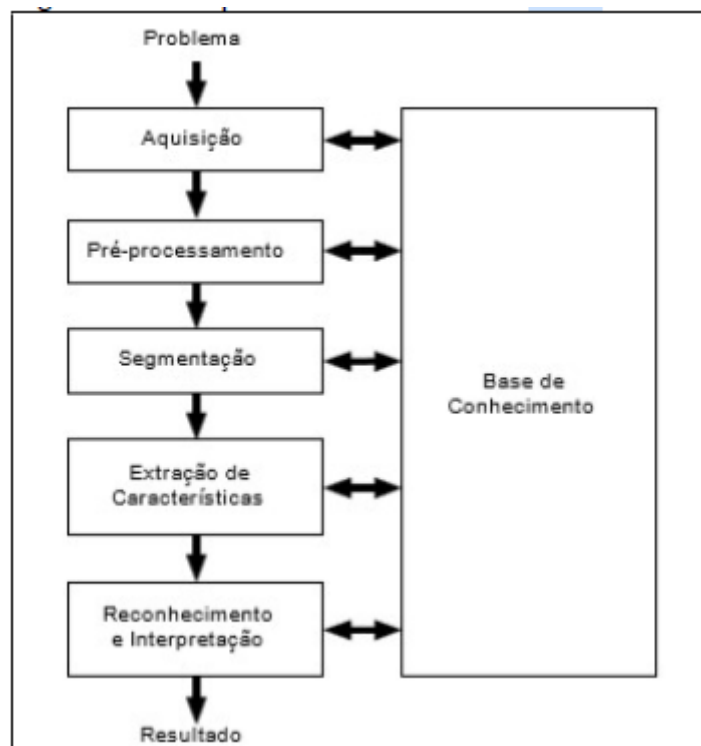


Figura 2. Etapas de um sistema de visão computacional

O pré-processamento é responsável por melhorar a imagem com o domínio do problema para obter um melhor resultado nas etapas seguintes, por exemplo, a aplicação de contraste ou isolamento de regiões (GONZALEZ; WOODS, 2000).

Na etapa de segmentação o seu objetivo é dividir a imagem em diferentes regiões de acordo com o problema, estas serão analisadas por algoritmos específicos para a extração de informação. Podemos citar como exemplo em uma imagem monocromática (preto e

branco), aplicar uma segmentação considerando os seus níveis de cinza ou suas diferenças (GONZALEZ; WOODS, 2000).

A representação e descrição são obtidas através dos resultados da segmentação em uma forma adequada para o processamento computacional que será realizado. Nesta etapa também ocorre uma extração de informações ou características de interesse (MARQUES FILHO; VIEIRA NETO, 1999).

A próxima etapa é composta por dois estágios, sendo o primeiro chamado de reconhecimento e a segunda de interpretação. Esta etapa é responsável por classificar os objetos a partir das informações extraídas da imagem juntamente com a base de conhecimento já estabelecida anteriormente e também atribuição de um significado a eles (GONZALEZ; WOODS, 2000).

A etapa com maior complexidade é base de conhecimento e também a mais importante, sua função é realizar a comunicação entre os módulos durante os processos e também guia-los (MARQUES FILHO; VIEIRA NETO, 1999).

O resultado final pode ser obtido em qualquer etapa ilustrada na figura 6, irá depender da complexidade de iterações da aplicação. Em alguns casos nem todas as etapas são utilizadas. A última e não menos importante é obter a saída, podendo está ser um conjunto de informações ou até mesmo uma imagem (GONZALEZ; WOODS, 2000) .

### 3.2 OpenCV

A biblioteca *Open Source Vision Library* (OpenCV), desenvolvida pela Intel em 2000, é um conjunto de mais de 2.500 algoritmos de programação em código aberto, voltada para o desenvolvimento de aplicações em visão computacional. Proporciona utilidades como, manipulação de imagens, desenvolvimento de algoritmos de processamento de imagens, análise de movimentos, reconhecimento de objetos, entre outros (CUNHA, 2013).

Desenvolvida nas linguagens C e C++, sendo compatível com os principais sistemas operacionais, Linux, Windows e Mac OS X. Projetada com o objetivo de tornar a visão computacional mais aberto a usuários e também desenvolvedores nas áreas de comunicação entre humano e computador em tempo real (BRADSKI; KAEHLER, 2008, tradução nossa).

A construção do OpenCV surgiu durante uma pesquisa da Intel para aplicações de intenso uso da CPU. Um dos pesquisadores da Intel notou que nas universidades como o MIT possuíam grupos de alunos com uma estrutura de visão computacional bem desenvolvida, e os alunos davam continuidade aos seus projetos reaproveitando o código já existente e os aperfeiçoando. A partir deste método, o OpenCV foi criado para tornar esta estrutura de visão computacional universalmente disponível (BRADSKI; KAEHLER, 2008, tradução nossa).

### 3.3 Contornos

Em processamento de imagens a definição de contorno pode ser dada como uma curva que liga todos os pontos contínuos, com mesma cor e intensidade é de grande utilidade na análise de formas e detecção e reconhecimento de objetos.

A biblioteca OpenCV possui duas funções para aplicação de contornos em imagens, são elas: *cv2.findContours* e *cv2.drawContours* . A função *cv2.findContours* quando aplicada é capaz de detectar os diferentes segmentos da imagem através de suas bordas e contornos. Já a *cv2.drawContours*, é responsável por desenhar os contornos e também pode ser utilizada para desenhar qualquer forma, através de métodos aplicados utilizando a função em questão (ROSEBROCK, 2016, tradução nossa).

A figura 3 exemplifica o método `cv2.findContours` em ação. Na parte superior do exemplo temos uma imagem que contém regiões brancas em um fundo escuro. Na parte inferior da ilustração temos a mesma imagem, com os contornos que o método irá encontrar. Os contornos podem ser rotulados com `cX` ou `hX`, onde a letra `c` significa *contour*, ou seja, contorno, e `h` significa *hole*, ou seja, buraco, e `X` é um número (BRADSKI; KAEHLER, 2008, tradução nossa).

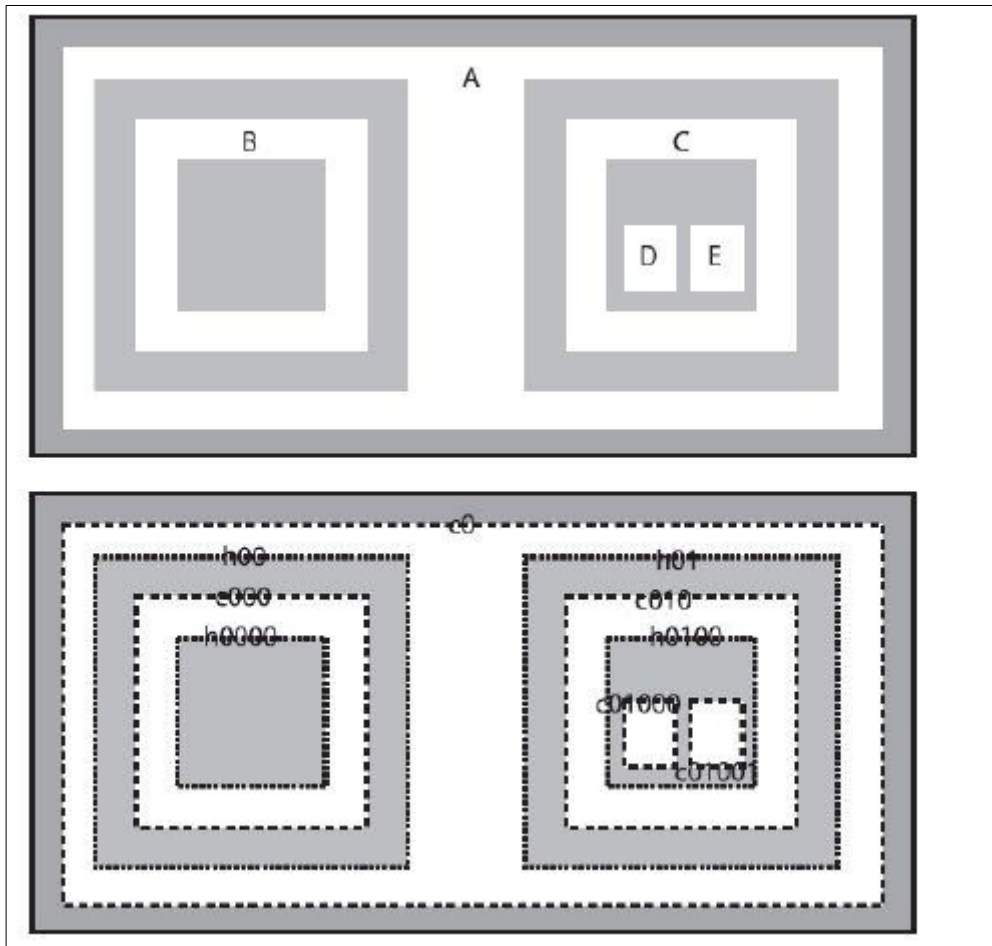


Figura 3 – Representação de uma imagem após utilizado o método : `cvFindContours()`

### 3.4 Laplace

O propósito da utilização do método de Laplace é a aplicação de bordas para determinar os contornos dos objetos presentes na imagem. Esta função pertence ao conjunto de algoritmos que compõem a biblioteca OpenCV através do comando `cv.Laplacian(src, dst, ddepth, ksize)`, onde `src` e `dst` são as imagens de origem e destino, `ddepth` é um código que indica a quantização da imagem resultante e `ksize` é o tamanho da máscara (GONZALEZ, WOODS, 2007).

O método torna-se eficiente devido à definição do próprio operador Laplaciano, que é utilizado para calcular a segunda derivada da imagem. Diferentemente dos filtros de primeira ordem que detectam as arestas com base no máximo e mínimo, o Laplacian detecta as arestas em cruzamento em zero, ou seja, onde a alteração de valor negativo para positivo ou vice-versa. A aplicação do método obtém-se como resultado o destaque de regiões de uma imagem que contém mudanças rápidas de intensidade, assim como outros operadores matemáticos como *Sobel* e *Scharr*, o Laplace é frequentemente utilizado para a detecção de bordas (ROSEBROCK, 2016, tradução nossa).

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

Figura 4 – Representação matemática do método de Laplace

A utilização de filtros juntamente como o método (figura 4) é necessária, estes aplicam a suavização dos ruídos ou diferenças indesejáveis de gradiente devido à maior sensibilidade ajustada de acordo com o tamanho do *Kernel*, sendo definido entre um e cinco, quanto maior, maior será a sensibilidade de detecção de bordas devido ao maior processamento de *pixels* (SINGH, KANG, 2019, tradução nossa).

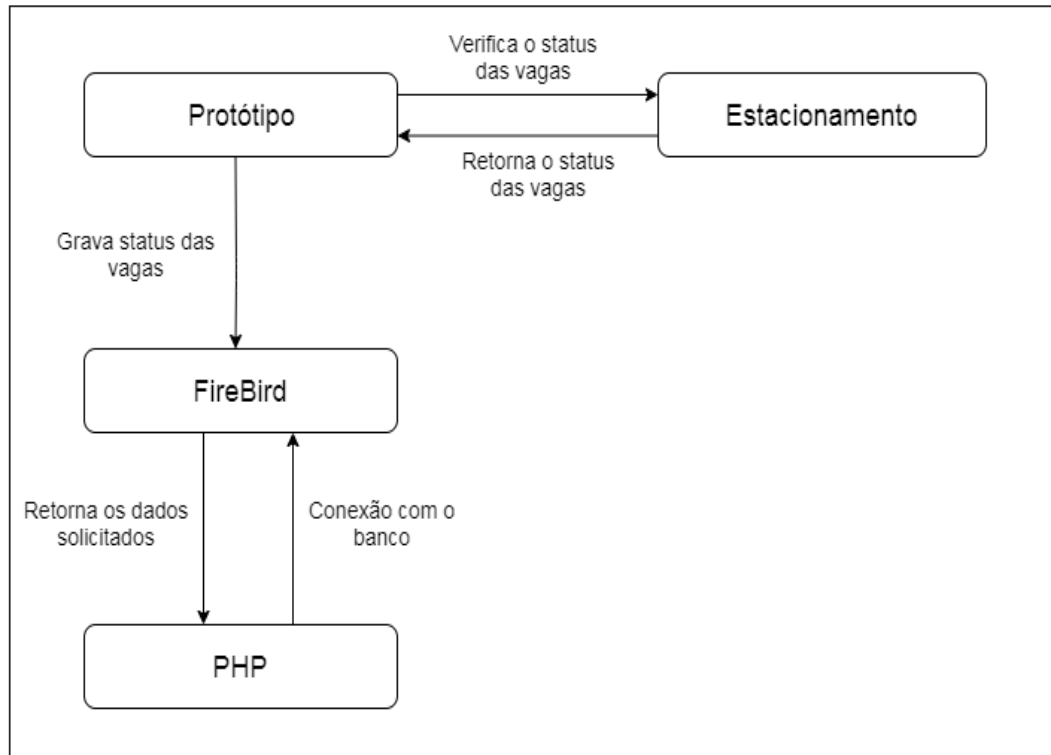
#### 4. Metodologia

Para o desenvolvimento deste projeto foram realizadas algumas etapas, a primeira etapa em questão foram abordados temas como: mobilidade urbana e o crescimento desordenado de automóveis nas vias e as consequências resultantes, dando maior ênfase na dificuldade de encontrar uma vaga de estacionamento, sendo ela em locais públicos ou privados; métodos de desenvolvimento utilizando a visão computacional e a biblioteca OpenCV para a identificação de objetos e aplicação de filtros.

As demais etapas são constituídas no desenvolvimento do protótipo, iniciando-se primeiramente pelo levantamento de recursos necessário, em seguida a instalação da câmera de monitoramento no estacionamento a ser aplicado o software, após isso foi dado início a programação para identificação das vagas e objetos nas áreas demarcadas, seguido da configuração do banco de dados e do servidor para visualização dos dados na página *WEB* e por fim os testes de reconhecimento das vagas ocupadas e livres.

##### 4.1. Fluxo dos dados e modelo de comunicação

Para o início do desenvolvimento, primeiramente estipulou-se um diagrama de fluxo de dados (figura 5), tendo como objetivo o mapeamento de todos os processos que o projeto é contido e todas as interconexões necessárias entre os mesmos. Com isso, facilitando uma visão global, facilitando o entendimento de todas as etapas a serem realizadas.



**Figura 5 - Fluxo de dados do projeto**

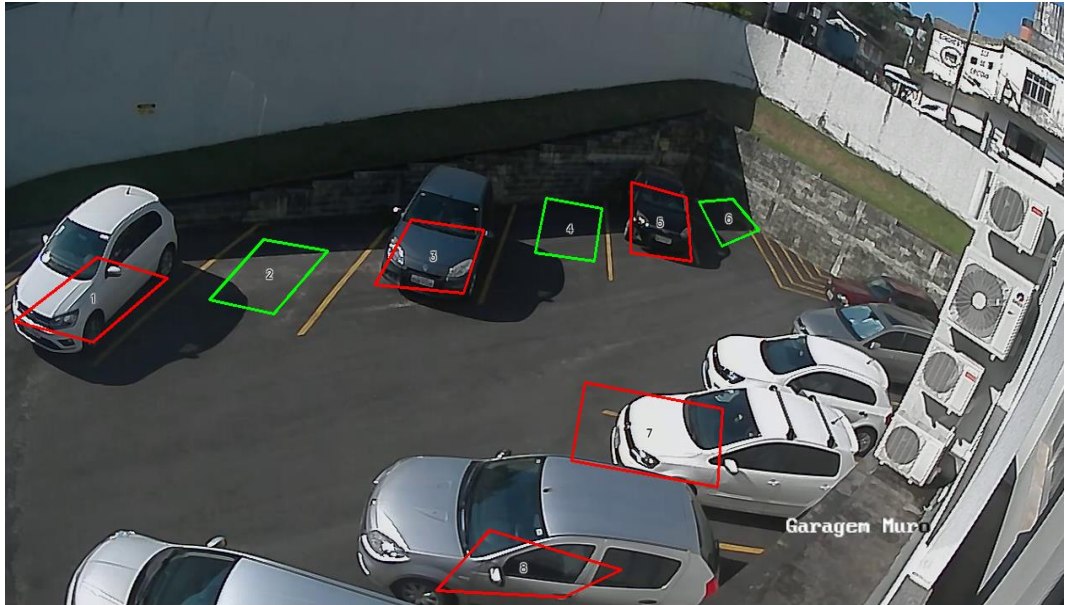
O fluxo inicial estipulado acontece no protótipo, onde este deve verificar através da câmera IP o status de cada vaga do estacionamento e realizar o envio para o banco de dados. Sendo que, esta conexão é intermediada por uma DLL localizada no protótipo, tendo esta o objetivo de centralizar todas as operações que envolvem de alguma forma o uso da base de dados.

Por fim, a aplicação é a última parte do fluxo, ao qual nesta será possível realizar o gerenciamento das informações coletadas. Onde para a conexão com a base de dados, segue o mesmo formato citado anteriormente via arquivo PHP.

#### **4.2. Desenvolvimento do protótipo**

Para a montagem do projeto, a etapa a seguir foi pesquisar e realizar testes para se familiarizar com a linguagem Python e a biblioteca OpenCV, desenvolvendo uma interface onde seria visualizado o vídeo capturado pela câmera em tempo real, em seguida o código para identificação das vagas. No decorrer desta etapa surgiu o maior desafio do projeto, o tratamento de imagens. A uma complexidade do OpenCV na teoria com na prática, mesmo possuindo diversas comunidades de discussão e uma vasta documentação, tornando mais difícil encontrar um caminho para iniciar o desenvolvimento do software cumprindo os objetivos do projeto.

Na figura 6 observa-se o funcionamento do software, baseia-se nos pontos definidos para aplicação dos métodos descritos anteriormente, a função utilizada para realizar a interligação dos pontos das vagas é a drawContours. Também pode ser utilizada para desenhar qualquer forma, desde que tenha definido os pontos de limite, a aplicação deste método se baseia em três argumentos principais, o primeiro é a captura da imagem de origem, o segundo argumento é a leitura dos pontos definido em uma lista, o terceiro é o índice dos contornos importante quando se deseja desenhar contornos individuais e os demais argumentos se compõem da escala da cor desejada para a criação do retângulo, e também a espessura do mesmo.



**Figura 6 – Representação das vagas com os pontos definidos**

Definido os pontos de cada vaga e os desenhado no vídeo, a etapa seguinte foi aplicar filtros a fim de poder identificar a alteração nas vagas demarcadas.

Dentre os métodos utilizados, o principal deles foi o método laplaciano, este filtro detecta bordas de objetos na imagem, que produz bordas finas da largura de um *pixel*, a imagem 7, ilustra a aplicação do método. Este método age em tempo real, juntamente com a captura do vídeo ao executar o software, em cada vaga são capturados os *frames* e aplicado a função *cv.Laplacian* que é executada a todo momento. Com isto, *frame a frame* é aplicado o filtro Laplace em tempo real na imagem capturada pela câmera.



**Figura 7 – Detecção de bordas com o método Laplace**

Esta função é composta por métodos matemáticos capazes de identificar mudanças nos *pixels* da imagem. Para realizar a aplicação da função, são necessários dois argumentos, primeiro é a imagem original em *grayscale* e no segundo argumento é informado

o tipo de dado da imagem final, neste projeto foi utilizado sessenta e quatro *bits*, para que todos os contornos da imagem sejam detectados.

### 4.3. Desenvolvimento da página WEB

Após o término de todo o desenvolvimento para identificação das vagas e alteração do status da mesma, deu-se início ao desenvolvimento da página web responsável por disponibilizar aos usuários a visualização das vagas livres e ocupadas.

Nas pesquisas realizadas sobre dispositivos móveis, percebeu-se que a utilização de *smartphones* é uma ferramenta muito útil e bastante utilizada no dia-a-dia das pessoas, e que todos possuem navegadores web, esse foi o principal motivo pela escolha para o desenvolvimento em uma página web, pois não se faz necessário reescrever todo o código fonte da mesma para diferentes sistemas operacionais que existem entres os *smartphones*.

A linguagem de programação escolhida nesta etapa foi a PHP com o objetivo para o desenvolvimento da parte lógica da página web. Para a criação de layout e demais funções, utilizado no projeto, foi utilizado o *framework* Angular, sendo este, uma plataforma utilizada para a construção de interfaces de aplicações desenvolvida pela Google, ao qual utiliza de tecnologias como *HTML*, *CSS* e *JavaScript*.

Com a linguagem definida, o próximo passo foi pensar de que maneira os usuários consigam visualizar as vagas e identifica-las com clareza, foi criado uma imagem da área do estacionamento escolhido, na mesma foi feita a marcação com números, como ilustra o apêndice C.

A figura 8 ilustra o layout da página web, onde cada vaga está representada em uma linha, as linhas possuem o número da vaga, status de livre ou ocupado e sua localidade caso esteja livre.

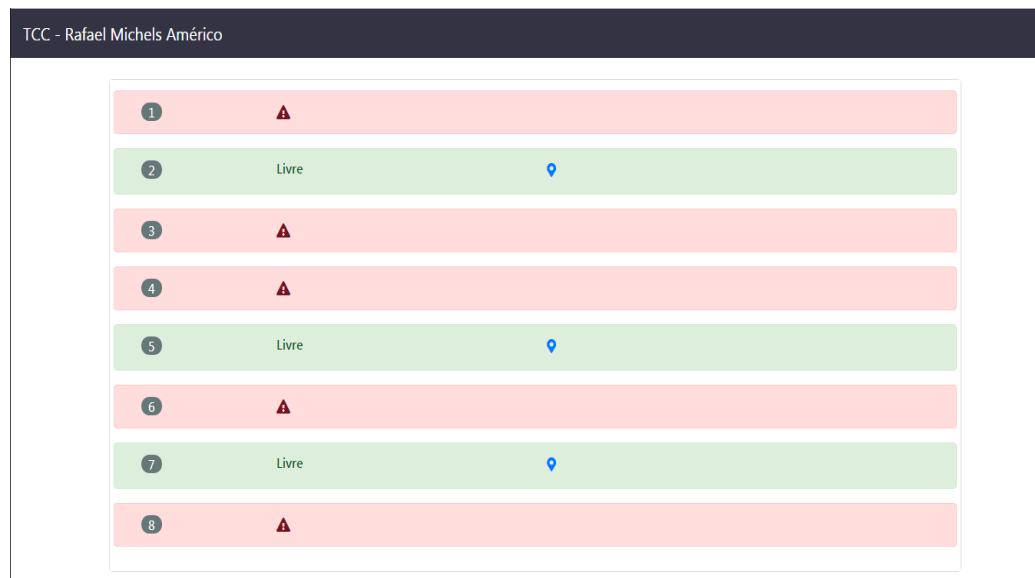


Figura 8 – Layout da aplicação WEB

### 4.4. Funcionamento da aplicação

Na figura 8 observa-se as informações disponíveis aos usuários, as vaga são representadas por um código apresentado no início de cada linha, seguida do *status*, que de acordo com o seu

valor altera a cor da linha para verde e apresentado o texto livre para as vagas disponíveis e também o ícone de localidade que mostra ao usuário qual a vaga o mesmo pretende se deslocar.

As linhas em vermelho representam as vagas ocupadas, descrito anteriormente todas as linhas iniciam com um código que representam o número de cada vaga, o *status* da vaga ocupada é representada por um ícone de atenção.

## 5. Conclusão

Com o término do projeto, além de proporcionar aquisição de conhecimento relacionado a tecnologias utilizadas para a realização do protótipo, também forneceu noções básicas referentes aos conceitos relacionados à visão computacional e processamento de imagens, as quais podem ser aplicadas em diversas áreas, como em realidade virtual, identificação de objetos, reconhecimento facial, dentre outras. Estas tecnologias em conjunto com a criatividade possibilitam o desenvolvimento de inúmeras aplicações.

Com o conhecimento abstraído da biblioteca OpenCV utilizada neste projeto para auxiliar no processamento de imagens, e com sua vasta comunidade, possibilitou o desenvolvimento do protótipo proposto.

Durante o desenvolvimento do projeto, algumas dificuldades foram encontradas com relação à parte prática, dentre elas podem ser citadas a complexidade na identificação de vagas e alteração do *status* da mesma, ao qual se fez necessário extensas pesquisas e tentativas até que se descobriu qual método a ser utilizado. O método de Laplace composto por métodos matemáticos pertence ao conjunto de algoritmos da biblioteca OpenCV, tem como objetivo em sua aplicação determinar os contornos dos objetos presentes na imagem.

Relativo ao conhecimento adquirido e etapas concluídas, foi possível o desenvolvimento de um protótipo para a detecção de vagas de estacionamento através de imagens capturadas por uma câmera IP, aplicando o processamento de imagens com auxílio da biblioteca OpenCV, verificando o *status* se estão livres ou ocupadas, e também uma aplicação WEB para visualização das vagas.

Com base nos testes realizados, obteve-se um resultado satisfatório quanto à integração entre todas as tecnologias implementadas e em comparação com o projeto de Antonio Henrique Mexas onde seu objetivo foi à identificação de carros e vagas de estacionamento através de imagens estáticas de alta resolução, obtendo uma taxa de acertos de 98% na busca por veículos nas vagas de estacionamento e 89% por vagas sem veículos, levando-se em conta que as variáveis encontradas em um ambiente de testes em tempo real são mais complexas e imprevisíveis, obteve-se uma acurácia de 95%, pode-se concluir que o presente trabalho obteve um bom resultado após as validações, sendo assim, pode ser aplicado em um ambiente real.

De acordo com o conhecimento obtido e nos resultados adquiridos com o projeto, sugere-se como trabalhos futuros:

- d) aplicação de inteligência artificial para identificação de vagas e reconhecimento de veículos nas mesmas;

aplicar a api Google Maps na aplicação WEB para fornecer uma melhor identificação da vaga e deslocamento até a mesma;

## Referências

- COSTA, Filipe Antonio dalla; COSTA, Osmar Antonio dalla. O Bem-estar de Suínos como Estratégia para Agregação de Valor. In: X CONGRESSO NORDESTINO DE PRODUÇÃO ANIMAL, 10, 2015, Teresina. O Bem-estar de Suínos como Estratégia para Agregação de Valor. Teresina: Snpa, 2015. p. 1 - 3. Disponível em: <https://ainfo.cnptia.embrapa.br/digital/bitstream/item/143313/1/final8031.pdf>. Acesso em: 15 set. 2018.
- BARWINSKI, Luísa. A World Wide Web completa 20 anos, conheça como ela surgiu. 2009. Disponível em: <<https://www.tecmundo.com.br/historia/1778-a-world-wide-web-completa-20-anos-conheca-como-ela-surgiu.htm>> Acesso em: 05 out. 2018.
- BRUNELLI, Roberto. Template Matching Techniques in Computer Vision: Theory and Practice. 2000. Disponível em: <[https://www.researchgate.net/publication/272496016\\_Template\\_Matching\\_Techniques\\_in\\_Computer\\_Vision\\_Theory\\_and\\_Practice](https://www.researchgate.net/publication/272496016_Template_Matching_Techniques_in_Computer_Vision_Theory_and_Practice)>. Acesso em: 12 out 2019.
- CUNHA, André Luiz Barbosa Nunes da. Sistema Automático para Obtenção de Parâmetros do Tráfego Veicular a partir de Imagens de Vídeo usando OpenCv. 2013. 128 p. Curso de Escola de Engenharia de São Carlos, Universidade de São Paulo, São Paulo, 2013. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/18/18144/tde-19112013-165611/pt-br.php>>. Acesso em: 18 maio 2019.
- BORGES, Eder. A mobilidade urbana centrada no uso do automóvel. 2012. Disponível em: <<http://periodicos.uem.br/ojs/index.php/Percorso/article/view/24798>>. Acesso em: 01 jun. 2019.
- GONZALEZ, Rafael C.; WOODS, Richard E. Processamento De Imagens Digitais, São Paulo: Edgard Blücher, 2000. Disponível em: <<https://edoc.pub/processamento-digital-de-imagens-3ed-pdf-free.html>>. Acesso em: 23 ago. 2018.
- GRANGE, Sébastien; FONG, Terrence; BAUR, Charles. TLIB: a Real-time Computer Vision Library for HCI. Sydney: Swiss Federal Institute Of Technology, 2003. Disponível em: <[https://www.ri.cmu.edu/pub\\_files/pub4/grange\\_sebastien\\_2003\\_1/grange\\_sebastien\\_2003\\_1.pdf](https://www.ri.cmu.edu/pub_files/pub4/grange_sebastien_2003_1/grange_sebastien_2003_1.pdf)>. Acesso em: 02 nov. 2018.
- HAMILTON, Emily. Como o estacionamento grátis piora o congestionamento. 2017. Disponível em: <<https://caosplanejado.com/como-o-estacionamento-gratis-piora-o-congestionamento/>>. Acesso em: 10 set. 2018.
- MARQUES FILHO, Ogê; VIEIRA NETO, Hugo. Processamento Digital de Imagens. Rio de Janeiro: Brasport, 1999. 410 p. Disponível em: <<http://www.ogemarques.com/wp-content/uploads/2014/11/pdi99.pdf>>. Acesso em: 07 maio 2019.
- MILLER, Erick G. Learned. Introduction to Computer Vision. Amherst, 2011. 11 p.

Disponível em:

<[https://people.cs.umass.edu/~elm/Teaching/Docs/IntroCV\\_1\\_19\\_11.pdf](https://people.cs.umass.edu/~elm/Teaching/Docs/IntroCV_1_19_11.pdf)>. Acesso em: 11 nov. 2018.

MONTEIRO, Geovana; VACCARI, Rafaella. A tecnologia associada ao nosso dia a dia.

2016. Disponível em: <<http://portal.metodista.br/rpcom/mundo-corporativo/a-tecnologia-associada-ao-nosso-dia-a-dia>>. Acesso em: 01 jun. 2018.

NETO, Gerlane. Mobilidade Urbana. 2018. Disponível em:

<<https://www.polemicaparaiba.com.br/paraiba/mobilidade-urbana-ciclovia-e-uma-opcao-de-locomocao-em-cidades-com-muito-congestionamento>>. Acesso em: 29 jan. 2019.

RIBEIRO, Breno. Entenda a evolução dos smartphones e como a sociedade se comporta com novas tecnologias. 2018. Disponível em:

<<https://www.folhavoria.com.br/geral/noticia/05/2018/entenda-a-evolucao-dos-smartphones-e-como-a-sociedade-se-comporta-com-novas-tecnologias>>. Acesso em: 15 nov. 2018.

RODRIGUE, Jean Paul. Urban Mobility and its Evolution. 2017. Disponível em:

<[https://transportgeography.org/?page\\_id=4617](https://transportgeography.org/?page_id=4617)>. Acesso em: 18 out. 2018.

ROSEBROCK, Adrian. OpenCV center of contour. 2016. Disponível em:

<<https://www.pyimagesearch.com/2016/02/01/opencv-center-of-contour/>>. 06 out. 2018.

SINGH, Atul Krishna; KANG, Pankaj. Laplacian of Gaussian (LoG). 2019. Disponível em:

<<https://theailearner.com/tag/cv2-laplacian/>>. Acesso em: 09 mar. 2019.

SZELISKI, Richard. Computer Vision: Algorithms and Applications. 2010. 979 p. Disponível

em: <[http://szeliski.org/Book/drafts/SzeliskiBook\\_20100903\\_draft.pdf](http://szeliski.org/Book/drafts/SzeliskiBook_20100903_draft.pdf)>. Acesso em: 04 nov. 2018.

ZHOU, Huiyu; WU, Jiahua; ZHANG, Jianguo. Digital Image Processing. [S.I]: Book Boon, 2010. 71 p. Disponível em:

<<https://pdfs.semanticscholar.org/17e5/d11c00370444d84ef351515e536df0f9abf8.pdf>>. Acesso em: 06 out. 2018.