

ANO
2019

JULIO CEZAR RUTKE | UMA ABORDAGEM BASEADA EM VISÃO COMPUTACIONAL
COM INTERNET DAS COISAS PARA CONTAGEM DE PASSAGEIROS EM TRANSPORTE
PÚBLICO URBANO



UDESC

UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC
CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

DISSERTAÇÃO DE MESTRADO

**UMA ABORDAGEM BASEADA EM
VISÃO COMPUTACIONAL COM
INTERNET DAS COISAS PARA
CONTAGEM DE PASSAGEIROS EM
TRANSPORTE PÚBLICO URBANO**

JULIO CEZAR RUTKE

JOINVILLE, 2019

Neste trabalho de dissertação, apresenta-se uma abordagem baseada em Visão Computacional e apoiada nos conceitos de Internet das Coisas (IoT) para o provimento de informações sobre o número de passageiros em veículos do transporte público, especificamente ônibus, buscando oferecer dados sobre o número de passageiros em cada rota. A meta é oferecer uma solução de baixo custo e totalmente transparente.

Através de técnicas de detecção de pessoas, especificou-se um algoritmo que identifica as pessoas que entram/saem do ônibus e acompanha seu trajeto dentro do alcance da câmera. Quando a pessoa atravessa uma linha limítrofe, o sistema realiza a contagem. Desenvolveu-se, um módulo de Contagem para ser acoplado na porta do ônibus. O módulo de Contagem, após a contabilização transmite as informações processadas via rede sem fio para um módulo Agregador fixo no ônibus. O módulo Agregador é responsável por unificar as informações de todas as portas e enviá-las para uma rede externa ao ônibus.

Orientador: Janine Kniess

Joinville, 2019

UNIVERSIDADE DO ESTADO DE SANTA CATARINA - UDESC
CENTRO DE CIÊNCIAS TECNOLÓGICAS - CCT
MESTRADO EM COMPUTAÇÃO APLICADA

JULIO CEZAR RUTKE

**UMA ABORDAGEM BASEADA EM VISÃO COMPUTACIONAL COM
INTERNET DAS COISAS PARA CONTAGEM DE PASSAGEIROS EM
TRANSPORTE PÚBLICO URBANO**

JOINVILLE

2019

JULIO CEZAR RUTKE

**UMA ABORDAGEM BASEADA EM VISÃO COMPUTACIONAL COM
INTERNET DAS COISAS PARA CONTAGEM DE PASSAGEIROS EM
TRANSPORTE PÚBLICO URBANO**

Dissertação submetida ao Programa de Pós-Graduação em Computação Aplicada do Centro de Ciências Tecnológicas da Universidade do Estado de Santa Catarina, para a obtenção do grau de Mestre em Computação Aplicada.

Orientador: Dra. Janine Kniess

JOINVILLE

2019

**Ficha catalográfica elaborada pelo programa de geração automática da
Biblioteca Setorial do CCT/UDESC,
com os dados fornecidos pelo(a) autor(a)**

Rutke, Julio Cezar
UMA ABORDAGEM BASEADA EM VISÃO
COMPUTACIONAL COM INTERNET DAS COISAS PARA
CONTAGEM DE PASSAGEIROS EM TRANSPORTE
PÚBLICO URBANO / Julio Cezar Rutke. -- 2019.
83 p.

Orientadora: Janine Kniess
Dissertação (mestrado) -- Universidade do Estado de
Santa Catarina, Centro de Ciências Tecnológicas, Programa
de Pós-Graduação em Computação Aplicada, Joinville, 2019.

1. Contagem de Passageiros. 2. Mobilidade Urbana. 3.
Internet das Coisas. 4. Visão Computacional. I. Kniess,
Janine. II. Universidade do Estado de Santa Catarina, Centro
de Ciências Tecnológicas, Programa de Pós-Graduação em
Computação Aplicada. III. Título.

**Abordagem Baseada em Visão Computacional com Internet das Coisas para
Contagem de Passageiros em Transporte Público Urbano**

por

Julio Cezar Rutke

Esta dissertação foi julgada adequada para obtenção do título de

Mestre em Computação Aplicada


Área de concentração em “Ciência da Computação”,
e aprovada em sua forma final pelo

**CURSO DE MESTRADO ACADÊMICO EM COMPUTAÇÃO APLICADA
DO CENTRO DE CIÊNCIAS TECNOLÓGICAS DA
UNIVERSIDADE DO ESTADO DE SANTA CATARINA.**

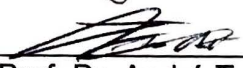
Banca Examinadora:



Profa. Dra. Janine Kniess
CCT/UDESC (Orientadora/Presidente)



Prof. Dr. William Alberto Cruz
Castañeda UTFPR



Prof. Dr. André Tavares da Silva
CCT/UDESC

Joinville,SC, 20 de fevereiro de 2019.

Dedico este trabalho aos meus familiares, amigos, colegas e professores que me acompanharam e me deram forças nessa magnífica trajetória. Mas principalmente a meu querido pai, que infelizmente não poderá ver mais essa conquista minha, porém eu sei que estaria orgulhoso.

AGRADECIMENTOS

Agradeço a Deus em primeiro lugar, pois nele nos movemos e nele vivemos, ele proporciona que a cada dia eu possa levantar com saúde e disposição para cumprir minha jornada. Ele moldou meus caminhos para que mesmo que antecipado eu chegasse a essa etapa e passasse com esmero.

Agradeço a minha família por ser a minha base e me sustentar a cada queda e recaída. Principalmente a minha irmã Janita e meu cunhado Adriano por me abrirem sua casa para que eu pudesse ter um teto, um abrigo, e assim me dedicar plenamente às minhas tarefas.

Agradeço aos meus professores pelos conhecimentos passados, proporcionando meu aprimoramento científico e acadêmico, em especial à minha orientadora, Dra. Janine Kniess, que teve paciência comigo, me ajudando em muitos momentos e contribuindo muito para que esse trabalho pudesse ser concluído com êxito.

Agradeço a empresa Transtusa pela parceria, por não medir esforços e me permitir utilizar seus ônibus quando houve a necessidade. Por colocar seus funcionários a disposição para apoio.

Agradeço aos meus colegas de laboratório que me proporcionaram momentos de debates, risos, reflexões e descontrações ajudando a relaxar a mente para que os trabalhos pudessem fluir mais levemente.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001. Portanto fica aqui também meu agradecimento à CAPES, que viabilizou esse processo e mais essa conquista em minha vida.

Enfim, a todos aqueles que de alguma forma me auxiliaram, direta ou indiretamente a ser quem eu sou hoje, torcendo, dando conselhos, ou me apoiando em momentos tristes e de desânimo.

Muito Obrigado!

“Deus julgará o justo e o ímpio; porque há um tempo para todo o propósito e para toda a obra.”

Eclesiastes 3.17

RESUMO

Um transporte público urbano eficiente contribui para a melhora da mobilidade urbana e consequentemente para a qualidade de vida dos cidadãos. A carência de informações sobre o número de passageiros que trafegam em cada uma das rotas disponibilizadas pelas empresas concessionadas, ou sobre a localização dos veículos em referência aos pontos de parada, é um dos elementos que dificulta o oferecimento de qualidade no transporte público.

Neste trabalho de dissertação, apresenta-se uma abordagem baseada em Visão Computacional e apoiada nos conceitos de Internet das Coisas (IoT) para o provimento de informações sobre o número de passageiros em veículos do transporte público, especificamente ônibus, buscando oferecer dados sobre o número de passageiros em cada rota. A meta é oferecer uma solução de baixo custo e totalmente transparente.

Através de técnicas de detecção de pessoas, especificou-se um algoritmo que identifica as pessoas que entram/saem do ônibus e acompanha seu trajeto dentro do alcance da câmera. Quando a pessoa atravessa uma linha limítrofe, o sistema realiza a contagem. Desenvolveu-se, um módulo de Contagem para ser acoplado na porta do ônibus. O módulo de Contagem, após a contabilização transmite as informações processadas via rede sem fio para um módulo Agregador fixo no ônibus. O módulo Agregador é responsável por unificar as informações de todas as portas e enviá-las para uma rede externa ao ônibus.

Com o objetivo de obter melhor precisão na contagem de passageiros, associou-se o algoritmo de contagem com as técnicas de Visão Computacional, Haar Likes e HOG. Além disso, avaliou-se o potencial do algoritmo com a técnica de *Deep Learning*, YOLOV3.

Utilizou-se para a implementação do sistema, microcontroladores do tipo Raspberry Pi e NodeMCU-ESP32, embarcados no ônibus. Nos experimentos, a captura dos dados foi realizada com câmeras de baixo custo. Testes de transmissão com as tecnologias de comunicação 802.11 b/g/n e GPRS foram realizados, visando avaliar o potencial destas tecnologias no envio dos dados no ônibus entre os microcontroladores e também para uma rede externa.

Palavras-chaves: Contagem de Passageiros. Mobilidade Urbana. Internet das Coisas. Visão Computacional.

ABSTRACT

Efficient urban public transport contributes to the improvement of urban mobility and consequently to the quality of life of citizens. The lack of information about the number of passengers, or vehicles location, is an element that becomes difficult to offer quality in public transport.

In this work, an approach based on Computer Vision and supported by concepts of Internet of Things (IoT) is presented to provide information about the number of passengers in each route in public transport vehicles, specifically buses. The goal is to offer a low-cost and fully transparent solution for transport users.

Through people-detection techniques, a passenger counting algorithm has been developed in order to identify who enter/leave in the bus, and follow their route while in the camera range. When one person crosses a boundary line, the system performs the count. It developed a counting module to be attached to the bus door. The Count module, after counting, transmits the processed information via wireless network to an aggregator module fixed in the bus. The Aggregator module is responsible for unifying information from all ports and sending them to a network outside the bus.

In order to obtain better accuracy in the passenger count, the counting algorithm was associated with the techniques of Computational Vision, Haar Likes e HOG. In addition, the potential of the algorithm was evaluated with the Deep Learning technique, YOLOV3.

For the system implementation, Raspberry Pi and NodeMCU-ESP32 microcontrollers were used in the bus. In the experiments, data capture was performed with low cost cameras. Transmission tests with the 802.11 b/g/n and GPRS communication technologies were carried out, aiming to evaluate the potential of these technologies in sending the data on the bus between the microcontrollers and also to an external network.

Key-words: Passenger counting. Urban Mobility. Internet of Things. Computational Vision.

LISTA DE ILUSTRAÇÕES

Figura 1.1 – Funcionamento do Sistema de Bilhetagem Integrado, Fonte: Próprio Autor.	17
Figura 2.1 – Processo de Características Haar, (Fonte: Adaptado de (VIOLA; JONES, 2001)).	27
Figura 2.2 – Processo de HOG (Adaptado (DALAL; TRIGGS, 2005)).	29
Figura 3.1 – Taxonomia de Sistemas para Contagem de Passageiros, Fonte: Próprio Autor.	35
Figura 4.1 – Arquitetura Lógica do Sistema, Fonte: Próprio Autor.	41
Figura 4.2 – Arquitetura de Implementação do Sistema, Fonte: Próprio Autor.	43
Figura 4.3 – Fluxograma do Algoritmo de Detecção e Contagem, Fonte: Próprio Autor.	44
Figura 4.4 – Fluxograma do Processamento do Vídeo no Algoritmo de Contagem, Fonte: Próprio Autor.	49
Figura 4.5 – Estrutura de Quadros de um Vídeo e a Detecção de Pessoas, Fonte: Próprio Autor.	50
Figura 4.6 – Fluxograma do Processamento de Quadros no Algoritmo de Contagem, Fonte: Próprio Autor.	50
Figura 4.7 – Comparação de Contagem, Fonte: Próprio Autor.	51
Figura 4.8 – Algoritmo de Agregação de Contagem das Portas, Fonte: Próprio Autor. .	53
Figura 4.9 – String Algoritmo de Agregação de Contagem das Portas, Fonte: Próprio Autor.	53
Figura 4.10 – Agregação da String de Contagem das Portas, Fonte: Próprio Autor. . . .	54
Figura 5.1 – Câmera na Posição com Ângulo de 45° Dentro do Ônibus, Fonte: Próprio Autor	58
Figura 5.2 – Câmera na Posição Zenital Dentro do Ônibus, Fonte: Próprio Autor .	59
Figura 5.3 – Contagem no Embarque com Câmera em Posição Zenital e Haar .	61
Figura 5.4 – Contagem no Desembarque com Câmera em Posição Zenital e Haar	62
Figura 5.5 – Contagem no Embarque com Câmera em Posição 45° e Haar . . .	63
Figura 5.6 – Contagem no Desembarque com Câmera em Posição 45° e Haar .	64
Figura 5.7 – Resultados da Contagem no Embarque com Câmera em Posição Zenital e HOG	65
Figura 5.8 – Resultados HOG no Desembarque com Câmera em Posição Zenital	66
Figura 5.9 – Resultados da Contagem no Embarque com Câmera em Posição 45° e HOG	67
Figura 5.10 – Contagem no Embarque com Câmera em Posição Zenital e YOLOV3	68
Figura 5.11 – Contagem no Desembarque com Câmera em Posição Zenital e YOLOV3	69
Figura 5.12 – Análise do Atraso Fim-a-Fim - Contagem e Agregador.	71

Figura 5.13—Análise da Perda de Pacotes - Contagem e Agregador.	72
Figura 5.14—Análise do Atraso Fim-a-Fim - Agregador e Servidor.	72
Figura 5.15—Análise da Perda de Pacotes - Agregador e Servidor.	73
Figura A.1—(a) Hardware do Módulo de Contagem e (b) Módulo de Agregação .	83

LISTA DE TABELAS

Tabela 5.1 – Classificação dos Vídeos Coletados	59
Tabela 5.2 – Resultados da Contagem (com Haar)	60
Tabela 5.3 – Resultados da Contagem (com HOG)	65
Tabela 5.4 – Resultados da Contagem (com YOLOV3)	67

LISTA DE SIGLAS E ABREVIATURAS

HOG Histogramas de Gradientes Orientados

SIT Sistema Integrado de Transporte

IBGE Instituto Brasileiro de Geografia e Estatística

Transtusa Transporte e Turismo Santo Antônio LTDA

GPS Sistema de Posicionamento Global

SVM Máquinas de Vetores de Suporte

LLNs Low Power and Lossy Networks

LPWANs Low Power Wide Area Networks

IDE Ambiente de Desenvolvimento Integrado

6LoWPAN IPv6 over Low power Wireless Personal Area Networks

CoAP Constrained Application Protocol

MQTT Message Queue Telemetry Transport

XML Extensible Markup Language

RGB Red, Green e Blue

YOLO You Only Look Once

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Contexto	17
1.2	Definição do Problema	18
1.3	Objetivos	20
1.4	Organização do Texto	21
2	FUNDAMENTAÇÃO TEÓRICA	22
2.1	Internet da Coisas	22
2.1.1	Protocolos de Comunicação para Internet das Coisas	24
2.2	Visão Computacional	26
2.2.1	Técnicas para Detecção de Objetos	26
2.2.1.1	<i>Características de Haar</i>	27
2.2.1.2	<i>Histogramas de Gradientes Orientados</i>	29
2.2.1.3	<i>Classificação</i>	30
2.2.1.4	<i>Deep Learning</i>	32
2.3	Conclusões do Capítulo	32
3	TRABALHOS RELACIONADOS	34
3.1	Contagem de Passageiros em Transporte Urbano	34
3.1.1	Abordagem para Contagem de Passageiros com Câmeras	36
3.2	Conclusões do Capítulo	39
4	ABORDAGEM DESENVOLVIDA	41
4.1	Arquitetura do Sistema	41
4.2	Algoritmo de Contagem	44
4.3	Algoritmo de Agregação	51
4.4	Conclusões do Capítulo	55
5	EXPERIMENTOS E RESULTADOS	57
5.1	Ambiente de Testes	57
5.2	Experimentos e Análise de Resultados	58
5.2.1	Análise do Algoritmo de Contagem de Passageiros	58
5.2.1.1	<i>Captura de Vídeos</i>	58
5.2.2	Resultados com o Algoritmo de Contagem	60
5.2.2.1	<i>Resultados com o Algoritmo de Contagem: HOG</i>	64
5.2.2.2	<i>Resultados com o Algoritmo de Contagem: YOLOV3</i>	67

5.2.3	Análise do Algoritmo de Agregação de Dados	68
5.3	Conclusões do Capítulo	73
6	CONCLUSÕES FINAIS	75
6.1	Trabalhos Futuros	77
	REFERÊNCIAS	78
	ANEXO A – REQUISITOS DE IMPLEMENTAÇÃO.	83

1 INTRODUÇÃO

Segundo (NATIONS, 2014), a maior parte da população atualmente vive na zona urbana. Esse número vem crescendo e a estimativa é que em 2050 o número de pessoas vivendo na zona urbana seja quase o dobro da população rural. O crescimento desse êxodo, traz consigo problemas de recursos e infraestrutura. Tornar as cidades mais inteligentes, pode contribuir significativamente para melhorar a qualidade de vida dos cidadãos.

Uma cidade inteligente liga a infraestrutura física, a infraestrutura de Tecnologia da Informação (TI), a infraestrutura social e a infraestrutura de negócios para alavancar a inteligência coletiva da cidade (HARRISON et al., 2010). Cidades Inteligentes devem oferecer tecnologias que conectam as pessoas, informações e elementos da cidade para criar uma cidade sustentável com comércio competitivo e inovador (BAKICI; ALMIRALL; WAREHAM, 2013). Nas Cidades Inteligentes, pode-se monitorar e integrar as condições de todas as suas infraestruturas críticas, incluindo estradas, pontes, túneis, trilhos, metrô, aeroportos, portos marítimos, comunicações, água, energia ou grandes edifícios (HALL, 2000).

Para ser categorizada como inteligente, uma cidade deve atender a seis indicadores (GIFFINGER; GUDRUN, 2010): Economia Inteligente, que considera fatores que interferem na renda per capita da cidade, despesas públicas e taxa de desemprego; Pessoas Inteligentes, que considera os aspectos que interferem no capital social e humano, como formação escolar e competências; Governança Inteligente, que analisa os fatores que interferem na administração da cidade; Meio Ambiente Inteligente, que trata dos fatores que interferem na natureza, como emissão de dióxido de carbono; Vida Inteligente, que estuda os fatores que interferem na qualidade de vida das pessoas da cidade, como saúde, turismo e cultura; Por fim, Mobilidade Inteligente, que considera os fatores que interferem na mobilidade urbana, como por exemplo, transporte público.

Uma Mobilidade Inteligente é crucial para as Cidades Inteligentes, pois impacta em vários aspectos da qualidade de vida dos cidadãos. A mobilidade inteligente pode auxiliar na redução de engarrafamentos e ainda diminuir o número de acidentes de trânsito. Pesquisadores ainda não adotaram um padrão sobre mobilidade inteligente e existem definições distintas.

Mobilidade Inteligente, segundo (ALBINO; BERARDI; DANGELICO, 2015), é o uso de tecnologias da informação e comunicação nos transportes para melhorar o tráfego urbano. Para (CHUN; LEE, 2015), uma mobilidade inteligente se faz através do

uso de sistemas de tráfego atuais, formados de estradas, veículos e sistemas de sinalização, integrados a uma tecnologia inteligente. (LENZ; HEINRICHS, 2017) afirmam que a Mobilidade Inteligente auxilia as pessoas a encontrar e explorar opções de transporte disponíveis, além de possibilitar o melhoramento do planejamento de transporte e mobilidade nas cidades. No trabalho proposto por (BENEVOLO; DAMERI; D'AURIA, 2016), os autores mencionam que a Mobilidade Inteligente está focada em melhorar a qualidade de vida dos cidadãos, e para isso nem sempre se faz necessário o uso de tecnologias. No entanto, para que uma maturidade seja atingida neste âmbito, o uso das tecnologias podem contribuir no gerenciamento e integração das informações. Apesar das diferentes definições é consenso, que a mobilidade inteligente se alcança melhorando a qualidade da mobilidade urbana.

A mobilidade urbana, refere-se a facilidade com que um deslocamento pode ser feito até um determinado lugar, partindo de qualquer ponto de uma cidade (FREITAS; PLACIDES; ARAÚJO, 2015). Uma área urbana com um sistema de mobilidade deficiente pode sofrer uma série de prejuízos como, a redução de investimentos no local, dificuldades de acesso a serviços essenciais e problemas relacionados ao meio ambiente (VIANNA, 2013). Nas cidades, o oferecimento de um transporte público eficiente é fundamental para melhorar a qualidade da mobilidade urbana.

Um sistema de transporte público é geralmente formado por veículos como, ônibus e metrô. Segundo a Constituição de 1988, no Brasil, cabe a administração municipal organizar a prestação do transporte coletivo. A maioria das cidades opta pelo uso de ônibus, pois o custo de implementação de sistemas metroviários é aproximadamente 20 (vinte) vezes maior (CARVALHO, 2017). Os sistemas de transporte público são de modo geral planejados e geridos pelas administrações municipais, porém delegados a iniciativa privada (GOMIDE, 2006).

A falta de informações, tais como indisponibilidade de rotas e tempo estimado de chegada nos pontos de parada, podem resultar na má distribuição da frota e superlotação nos veículos. As rotas percorridas por uma frota de ônibus apresentam características diferentes ao longo do dia em nível de lotação. Em horários próximos ao início ou término de expedientes convencionais, normalmente o tráfego de pessoas aumenta. Neste contexto, informações sobre o número efetivo de pessoas que trafegam em cada rota pode proporcionar uma melhor distribuição dos ônibus em relação as rotas. Sistemas embarcados podem coletar informações de diversas fontes, tais como gerenciamento de tráfego, parque de estacionamento, superlotação e tempo de espera pelo transporte público. A implementação de sistemas dessa natureza torna-se desafiadora em ambientes de transporte público, que apresentam um grande fluxo de pessoas e número de terminais de embarque e desembarque.

Diversas empresas utilizam o sistema de bilhetagem eletrônica para realizar

a contagem de passageiros (BERNINI et al., 2014). O passageiro possui um número de créditos em um cartão dotado de *chip* e memória. Quando o usuário aproxima o cartão do validador (equipamento de leitura do cartão) que fica dentro do ônibus, o valor da passagem é descontado dos créditos armazenados, e a catraca (se existir) é então liberada. No modelo de bilhetagem eletrônica é importante observar que a informação sobre o número de passageiros por trajeto percorrido pode ser imprecisa. Isto ocorre, especialmente pelo fato da maioria das empresas adotarem um sistema de bilhetagem eletrônica integrado entre terminais.

A Figura 1.1 representa o esquema de funcionamento do sistema de bilhetagem integrado.

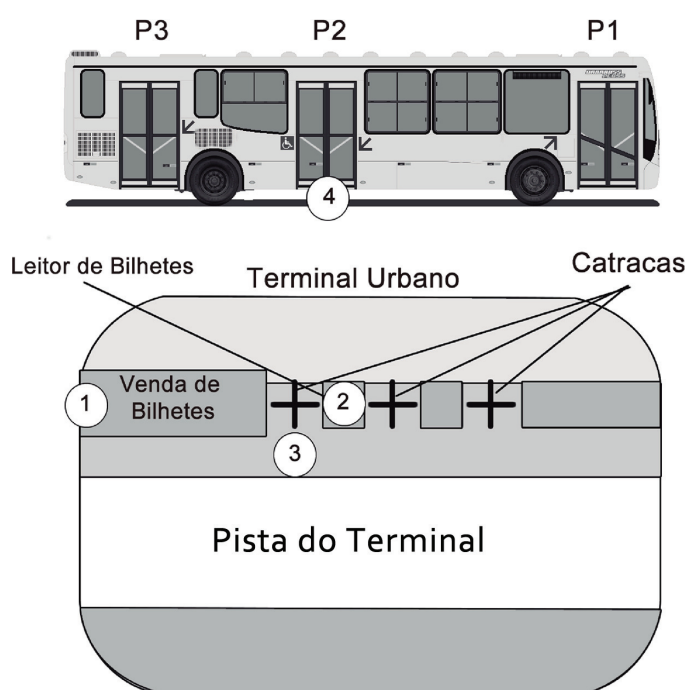


Figura 1.1 – Funcionamento do Sistema de Bilhetagem Integrado, Fonte: Próprio Autor.

1.1 CONTEXTO

O ambiente/modelo genérico de um sistema de bilhetagem eletrônica integrado, no Brasil, consiste dos seguintes elementos como descrito na Figura 1.1. Para que o usuário possa acessar um terminal urbano, ele necessita de um bilhete que pode ser adquirido fora do terminal (define-se terminais como estruturas físicas por onde os ônibus circulam, iniciando e/ou terminando uma determinada rota) (representado na Figura 1.1 pelo número 1). Após adquirir o bilhete, o passageiro deve registrá-lo em um leitor (número 2 na Figura 1.1) e acessar o terminal urbano passando por uma catraca (número 3 na Figura 1.1). A partir deste ponto, o passageiro pode entrar em qualquer um dos ônibus presentes no terminal pela(s) porta(s) traseira(s) e/ou centrais (quando

houver) (representadas por *P2* ou *P3* na Figura 1.1) sem a necessidade de registrar o bilhete novamente (número 4 na Figura 1.1). Adicionalmente, no sistema integrado, quando um ônibus chega a um novo terminal, o passageiro pode trocar de ônibus, acessando-o também pela(s) porta(s) traseira(s) e/ou centrais (*P2* ou *P3* na Figura 1.1) (quando houver) sem um novo pagamento. Durante a rota entre um terminal e outro, existem paradas de embarque/desembarque. Nessas paradas os passageiros descem pelas porta traseiras e/ou centrais (*P2* e *P3* na Figura 1.1) e embarcam pela porta frontal (*P1* na Figura 1.1). Ao entrar no ônibus através de uma parada intermediária entre terminais, o passageiro deve registrar um bilhete no leitor disponível no ônibus (*P1* na Figura 1.1).

O modelo representado pela Figura 1.1 é comum no Brasil, principalmente nas cidades de médio e grande porte porque facilita o trânsito de passageiros entre terminais com um número reduzido de bilhetes.

O transporte público urbano da cidade de Joinville, norte do estado de Santa Catarina, Brasil, será o ambiente de estudo de caso. Em Joinville, o transporte público através de ônibus é concessionado para empresas privadas e é dividido em terminais urbanos integrados. A cidade de Joinville possui o que é chamado de Sistema Integrado de Transporte (SIT), dividido em 10 terminais urbanos e funciona com desenho tronco alimentador, onde os passageiros embarcam em bairros nas linhas de pequeno porte, denominadas alimentadoras, e seguem até uma estação ou terminal, onde podem acessar uma linha de grande porte, chamadas de troncais ou estruturais. O sistema é operado por 257 linhas regulares, sendo 21 linhas troncais entre paradoras e diretas; 17 linhas interestações; 184 linhas alimentadoras e especiais; e 4 linhas vizinhas (SEPUD, 2017). A cidade de Joinville, segundo o censo do Instituto Brasileiro de Geografia e Estatística (IBGE) de 2010, possui 515.288 habitantes em uma densidade demográfica de 457,58 hab/km².

1.2 DEFINIÇÃO DO PROBLEMA

O gerenciamento eficiente da frota de ônibus no cenário alvo, depende principalmente de uma contagem precisa do número de passageiros que trafegam nas diferentes rotas e horários das linhas de ônibus oferecidas. No transporte urbano público de Joinville, a contagem é realizada através do número de passageiros obtidos com a bilhetagem, pelo motorista do ônibus, e eventualmente uma pessoa é alocada para tal serviço. A frota de ônibus que atende pelo transporte público é concessionada para duas empresas privadas, Gidion, responsável pela zona sul da cidade e Transporte e Turismo Santo Antônio LTDA (Transtusa), que atende a zona norte de Joinville.

A contagem de passageiros no cenário alvo deste trabalho (modelo de bilhetagem eletrônica com terminais integrados), envolve alguns desafios:

- Como realizar a contagem de passageiros de forma transparente? Em outras palavras, o sistema não deve exigir a participação ativa dos passageiros;
- Como detectar e contar as pessoas, considerando que os passageiros apresentam características diferentes como, tamanho e forma?;
- Como diferenciar objetos de pessoas? Os passageiros podem portar acessórios como malas e mochilas. Ainda, como diferenciar se o passageiro está realizando um embarque ou desembarque? Soma-se a esta última questão, o número significativo de pessoas que podem acessar o ônibus em horários de movimentação intensa.

Em face dos questionamentos apresentados, nesta dissertação foi realizada uma pesquisa científica buscando identificar os elementos de hardware e software que possibilitem atender as demandas listadas.

Atualmente, nos ônibus disponibilizados pela empresa de transporte Transtusa na cidade de Joinville não existe uma infraestrutura de rede para o acesso a Internet ou rede externa. A empresa dispõe de uma rede sem fio no pátio da empresa, onde os ônibus retornam após o percurso da rota para o abastecimento de combustível, limpeza e manutenções. Nos terminais, a empresa dispõe de uma rede sem fio para uso particular da empresa. Neste sentido, definiu-se que o sistema deve operar embarcado nos ônibus, realizando a contagem internamente no ônibus sem dependência de um processamento externo para este fim.

Dentre as tecnologias disponíveis para uso embarcado, verificou-se que as tecnologias de hardware no âmbito de Internet das Coisas (IoT) oferecem o suporte necessário, como por exemplo, microcontroladores, sensores, dentre outros. Além disso, oferece uma gama de protocolos que possibilitam a comunicação entre os componentes embarcados com uma rede externa para a transmissão dos dados da contagem.

No escopo da identificação e contagem de passageiros, constatou-se que a Visão Computacional disponibiliza tecnologias com capacidade de extrair informações úteis contidas em imagens. No ambiente representado pela Figura 1.1 (denominado aqui de cenário alvo), os passageiros podem transportar objetos de grande porte como, malas e caixas, inviabilizando o uso de sensores (por exemplo, sensores infravermelho), que não dispõem de mecanismos para diferenciar pessoas de objetos. Neste contexto, decidiu-se pelo uso de câmeras instaladas nos ônibus para capturar o embarque e desembarque de passageiros.

A aplicação de câmeras exige o uso de algoritmos adequados para a identificação das pessoas com suas diferentes formas e tamanhos, bem como, para a extração das informações relevantes da imagem, neste caso, a pessoa, ignorando os acessórios transportados pelas pessoas do processo de contagem.

Neste quadro, identificou-se que a viabilidade do sistema no cenário de interesse impõe algumas premissas:

- Deve existir embarcado no ônibus o hardware adequado para armazenamento e processamento das informações.
- Os equipamentos embarcados devem prover a capacidade de comunicarem entre si para a troca de informações sobre a contagem.
- Para transmissão dos dados de contagem para a Internet a empresa deve fornecer uma tecnologia de comunicação sem fio de longo alcance ou prover uma infraestrutura de rede sem fio.
- Deve existir no ônibus os dispositivos aptos para a detecção das pessoas.
- O ônibus deve prover fontes de alimentação de energia para os dispositivos embarcados.

O sistema deve atender aos seguintes requisitos funcionais:

- Indubitabilidade no processo de identificação dos passageiros, aferindo-se pessoas e desprezando-se outros objetos que entram ou saem do ônibus.
- Capacidade de adaptação a variações de luminosidade, trepidações do ônibus e outros ruídos nos equipamentos utilizados para a detecção e contagem.
- Os dispositivos instalados no ônibus devem possuir a capacidade de registrar quantas pessoas entram ou saem do ônibus de forma transparente para os passageiros.

1.3 OBJETIVOS

O objetivo deste trabalho é desenvolver e analisar o desempenho de uma solução para contagem de passageiros em transporte público urbano (ônibus) em cenários que utilizem terminais integrados e com sistema de bilhetagem eletrônica, utilizando Visão Computacional e Internet das Coisas.

Para tanto, são perseguidos os seguintes objetivos específicos:

- Implantar no ônibus uma infraestrutura embarcada de hardware para detecção e contagem de passageiros.
- Especificar e implementar um algoritmo para detecção e contagem de passageiros, com base em Visão Computacional.
- Especificar e implementar um mecanismo de agregação que integre o hardware embarcado no ônibus com a rede externa da empresa ou Internet, com base em Internet das Coisas;

1.4 ORGANIZAÇÃO DO TEXTO

Este texto está organizado na seguinte forma: no Capítulo 2, apresenta-se o embasamento teórico deste trabalho. Descreve-se, no Capítulo 3 os trabalhos relacionados aos sistemas para detecção e contagem de passageiros em transporte urbano coletivo. No Capítulo 4, descreve-se a modelagem e o desenvolvimento da solução para a contagem de passageiros e para a transmissão dos dados coletados no ônibus. No Capítulo 5, são apresentados os experimentos realizados e a análise de desempenho da abordagem proposta. Por fim, no Capítulo 6, são apresentadas as conclusões do trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Diversas tecnologias são utilizadas para a implementação de aplicações de Cidades Inteligentes. Dentre elas, destacam-se: Internet das Coisas, Computação em Nuvem e Big Data (KON; SANTANA, 2016).

Big data é um termo dado para um conjunto de técnicas de processamento de dados que aproveitam o poder de muitos computadores para executar tarefas em paralelo em conjuntos de dados muito grandes (NAMBIAR; SHROFF; HANDY, 2018). Computação em nuvem é o poder de armazenar e acessar dados e programas pela Internet, substituindo a utilização do disco rígido dos computadores. A computação em nuvem permite que você mantenha informações em um servidor remoto (a nuvem) (AGARWAL; AGARWAL, 2017).

A Internet das Coisas (do Inglês *Internet of Things* - IoT), refere-se à coleção de dispositivos que podem se conectar à Internet para se comunicar e trocar dados (NAMBIAR; SHROFF; HANDY, 2018). Através da IoT novas aplicações podem funcionar de forma transparente nos mais diversos domínios, ou seja, de forma automática sem a intervenção humana. Para a contagem de passageiros, objeto de pesquisa deste trabalho de mestrado, a utilização dos conceitos associados à Internet das Coisas é fundamental para o desenvolvimento do sistema. No escopo da abordagem proposta (ver Capítulo 4), serão implantados microcontroladores embarcados com capacidade de comunicação interna e externa ao ônibus.

Para que a comunicação possa ocorrer, dados precisam ser coletados e processados. Neste trabalho são coletadas imagens, e as mesmas são processadas para realizar a contagem. Esse processamento de contagem é realizado utilizando-se conceitos de Visão Computacional.

Dessa forma, neste capítulo são apresentados os fundamentos relevantes para a pesquisa desenvolvida. Na Seção 2.1 são descritos os conceitos sobre os componentes e as tecnologias de comunicação para Internet das Coisas. Por fim, na Seção 2.2 são apresentados conceitos de Visão Computacional, além das técnicas de detecção de objetos para identificação e contagem das pessoas.

2.1 INTERNET DA COISAS

O termo Internet das Coisas apareceu pela primeira vez em 1999 com Kevin Ashton (ASHTON, 2011), no contexto de uma apresentação sobre o uso de sistemas *Radio-frequency identification* (RFID) no rastreamento de objetos de uma cadeia de

produção. Esses sistemas são compostos por leitores e etiquetas que podem por meio de radiofrequência trocar informações. Através dos RFIDs é possível que objetos possuam identificadores únicos e outras informações, como modelo, cor e sua localização no estoque por exemplo. A disseminação e implantação dos conceitos de IoT, deve-se a integração de várias tecnologias como: RFIDs, Rede de Sensores Sem Fio (RSSF) e Middleware (ATZORI; IERA; MORABITO, 2010).

Sistemas RFID são compostos por leitores e etiquetas RFID (FINKELZELLER, 2003). As etiquetas são caracterizadas por um identificador único e são aplicadas a objetos, pessoas ou animais. Portanto, RFIDs podem ser usados em uma ampla variedade de cenários de aplicações, abrangendo desde logística, saúde e segurança. Uma etiqueta, ou tag, RFID é um pequeno microchip conectado a uma antena usada tanto para receber o sinal da leitora quanto para transmitir as informações. Geralmente são embaladas em cartões, pulseiras ou chaveiros.

Redes de Sensores Sem Fio (RSSF) consistem em um número de nós sensores que realizam comunicação sem fio e são capazes de coletar informações do ambiente em que se inserem como, localização, temperatura e movimento (AKYILDIZ et al., 2002). Os objetivos das redes de sensores são, a eficiência energética (recurso mais escasso na maioria dos cenários que envolvem redes de sensores), escalabilidade (o número de nós na rede pode ser muito alto), confiabilidade (a rede deve ser capaz de relatar eventos de alarme urgentes) e robustez (nós sensores estarão sujeitos a falhas por várias razões) (ATZORI; IERA; MORABITO, 2010).

O Middleware é uma camada de software ou um conjunto de subcamadas interpostas entre os níveis de hardware e de aplicação. Sua função principal é de ocultar os detalhes de diferentes tecnologias para isentar o programador de questões que não são diretamente pertinentes ao seu foco, que é o desenvolvimento da aplicação específica permitida pelas infraestruturas de IoT (ATZORI; IERA; MORABITO, 2010). As redes em IoT podem ser heterogêneas, composta por dispositivos diversos com arquiteturas distintas. *Middlewares* neste contexto auxiliam na integração entre os objetos possibilitando que a comunicação possa ser realizada com sucesso (ARASTEH et al., 2016).

Nas aplicações de Internet das Coisas é amplamente utilizada a comunicação sem fio que pode ser realizada através de diversas tecnologias de rede como: Low Power and Lossy Networks (LLNs), Low Power Wide Area Networks (LPWANs), redes celulares e redes locais sem fio. Na solução para contagem de passageiros (descrita no Capítulo 4), utilizou-se as redes celulares e as redes locais sem fio do tipo IEEE 802.11 b/g/n. A escolha dessas tecnologias, deve-se ao fato de que ambas oferecem vantagens para a comunicação no cenário alvo. As redes celulares possuem longo alcance. Dessa forma, os dados capturados sobre a contagem no ônibus podem facil-

mente serem enviados para uma rede externa. A tecnologia 802.11 b/g/n está amplamente difundida em microcontroladores e permite o envio de dados de curto alcance entre os microcontroladores no ônibus.

A transmissão dos dados utilizando protocolos tradicionais da internet como TCP/IP e HTTP não são adequados para serem utilizados em dispositivos de IoT, considerando que os dispositivos para Internet das Coisas possuem limitações em termos de memória, processamento e energia. Diversos protocolos de comunicação foram propostos para o contexto de Internet das Coisas. Dentre eles destacam-se: *IPv6 over Low power Wireless Personal Area Networks (6LoWPAN)*, *Constrained Application Protocol (CoAP)* e *Message Queue Telemetry Transport (MQTT)* (SHENG et al., 2013).

O protocolo 6LoWPAN é uma versão do IPv6, otimizada para aplicação em redes LoWPANs (KUSHALNAGAR; MONTENEGRO; SCHUMACHER, 2007), que por sua vez, são redes de comunicação de baixo custo que permitem a conectividade sem fio em aplicativos com potência e recursos limitados (KUMAR; TIWARI, 2012). A implementação do 6LoWPAN está disponível para a tecnologia de comunicação 802.15.4, utilizada por vários microcontroladores no contexto de IoT, por exemplo, XBee e Pic.

Na subseção 2.1.1, discute-se os protocolos que foram adotados no escopo deste trabalho, a saber os protocolos de Camada de Aplicação CoAP e MQTT.

2.1.1 Protocolos de Comunicação para Internet das Coisas

Redes de baixa capacidade podem apresentar problemas de desempenho ao utilizar os protocolos tradicionais como por exemplo, o HTTP. Nesse sentido, alguns protocolos foram desenvolvidos, como é o caso do CoAP e do MQTT (SHENG et al., 2013). Ambos os protocolos atuam na Camada de Aplicação e são mais leves que o HTTP.

O protocolo CoAP foi proposto para redes de baixa potência e com interferências. O CoAP foi projetado para a transferência habilitada para a web e é facilmente integrado com URLs de suporte HTTP e diferentes tipos de mídia. O CoAP se destaca no tratamento da latência da transferência, da sobrecarga de sinalização e da perda de pacotes (SHELBY; HARTKE; BORMANN, 2014). Ele usa o formato de dados EXI (Efficient XML Interchanges), que é um formato de dados binários, eficiente em termos de espaço em comparação ao texto simples HTML/XML (SETHI; SARANGI, 2017). Outros recursos são oferecidos: compactação de cabeçalho, descoberta de recurso, configuração automática, troca de mensagem assíncrona, controle de congestionamento e suporte a mensagens multicast (SETHI; SARANGI, 2017). Existem quatro tipos de mensagens no CoAP: Non confirmable, Confirmable, Reset e Acknowledgement (VIL-

LAVERDE et al., 2012). A transmissão é realizada via protocolo de transporte UDP e usa o DTLS (*Datagram Transport Layer Security*) para fins de segurança.

O MQTT é um protocolo desenvolvido pela IBM e executa sob protocolo de camada de Transporte TCP. A troca de mensagens acontece no padrão *publish/subscribe*. Os clientes são publicadores/assinantes e o servidor atua como um intermediário ou Broker ao qual os clientes se conectam por meio do TCP. Os clientes podem publicar ou assinar um tópico. Essa comunicação ocorre por meio do Broker, cujo trabalho é coordenar as assinaturas e também autenticar o cliente. Quando um cliente publica alguma mensagem em algum tópico, todos clientes que assinaram esse tópico irão receber a mensagem publicada. O MQTT é mais leve que o HTTP, o que o torna adequado para aplicações IoT (SETHI; SARANGI, 2017). O MQTT tem um cabeçalho fixo de 2 bytes e até 12 bytes de cabeçalhos variáveis.

Para garantir a confiabilidade da transmissão de mensagens, o MQTT suporta 3 níveis de Qualidade de Serviço (QoS) (BEHNEL; FIEGE; MUHL, 2006). No nível 0, o MQTT envia a mensagem uma única vez e não confere se a mensagem chegou em seu destino. No nível 1, envia a mensagem ao menos uma vez e confere se a mensagem chegou no destino através de um pacote *PUBACK*. Quando o *PUBACK* é perdido, o servidor irá reenviar a mensagem, uma vez que a origem não recebeu a confirmação de recebimento. No nível 2, garante a qualidade em um processo de 4 vias. Envia a mensagem para o destino e aguarda a confirmação. O destino envia um pacote *PUBREC* para confirmar o recebimento. Ao receber a confirmação, a origem envia um pacote *PUBREL* informando ao destino que a confirmação foi recebida. Por fim, o destino retorna para a origem um pacote *PUBCOMP*.

Na solução proposta no Capítulo 4, o CoAP foi utilizado para o envio de mensagens entre os microcontroladores dentro do ônibus, enquanto o MQTT de QoS nível 1 foi utilizado para a transmissão dos dados para um servidor externo ao ônibus. Os módulos de contagem localizados nas portas, após a contagem enviam os dados para o agregador utilizando CoAP. Com a agregação concluída, os dados são transmitidos para um servidor em uma rede externa ao ônibus, através da tecnologia de comunicação GPRS via MQTT.

A literatura sobre técnicas para a contagem de pessoas engloba o uso diferentes tipos de hardware, como por exemplo, sensores fotoelétricos, sensores de pressão e câmeras de vídeo (PÁDUA; CAMPOS; CARCERONI, 2003). Entretanto, as abordagens com sensores apresentam diversas limitações na contagem de pessoas como, alta sensibilidade a ruído, variações de temperatura, poeira e fumaça. Além disso, eles não podem distinguir entre um passageiro e um grupo de passageiros ou objetos de pessoas (YAHIAOUI; KHOUDOUR; MEURIE, 2010). As técnicas baseadas na análise de imagens de cena vem ganhando destaque por mostrarem resultados significati-

vos na identificação de pessoas em diferentes tipos de cenários (PÁDUA; CAMPOS; CARCERONI, 2003). Estas técnicas geralmente aplicam ferramentas e conceitos da Visão Computacional. Na seção a seguir são apresentadas algumas técnicas para a detecção de objetos e pessoas encontradas na literatura que se baseiam no uso de câmeras de vídeo.

2.2 VISÃO COMPUTACIONAL

Nesta seção são discutidas algumas técnicas de detecção de objetos e pessoas para o tratamento de imagens como fundamentação teórica deste trabalho. Nas Seções 2.2.1.1 e 2.2.1.2, apresentam-se os conceitos sobre as técnicas de Haar Likes e Histogramas de Gradientes Orientados e sua relação com o problema da contagem de passageiros. Por fim, discute-se o emprego de técnicas de *Deep Learning* na detecção de pessoas e contagem de passageiros.

2.2.1 Técnicas para Detecção de Objetos

Os seres humanos conseguem perceber o que está a sua volta de forma fácil através do sentido da visão. Através de um conhecimento prévio, as pessoas conseguem classificar os objetos que veem em segundos. Por meio da visão o ser humano consegue identificar em três dimensões formas, cores, pessoas, movimentos, etc. A Visão Computacional é o processo de replicação deste sentido humano, através de modelagens utilizando software e hardware. A Visão Computacional estuda formas de reconstruir, interromper e compreender cenas 3D a partir de imagens 2D.

Segundo (SZELISKI, 2010), a Visão Computacional permite rastrear pessoas em movimento contra fundos difíceis de serem distinguidos. Permite ainda identificar, mesmo que em menor precisão, pessoas em determinadas fotos utilizando padrões de rosto, roupas e cabelo. Além da identificação de pessoas, outras áreas vem sendo alvo da Visão Computacional, como: Reconhecimento óptico de caracteres (OCR), Inspeção de Máquinas, Predição Médica, Segurança em Automóveis, Vigilância e Reconhecimento de Digitais.

Uma das áreas que ganhou destaque na Visão Computacional é o reconhecimento de pessoas. O reconhecimento segundo (POPPE, 2010) pode ser obtido em vários níveis de abstração. (MOESLUND; HILTON; KRÜGER, 2006) propõe uma hierarquia que divide o reconhecimento em: Ação Primitiva, Ação e Atividade. Ação Primitiva seria o movimento de algum membro, como colocar a perna direita para frente. Uma Ação seria a sequência de Ações Primitivas, como exemplo, correr. Uma atividade seria uma sequência de Ações, por exemplo de uma corrida com obstáculos que é composta por corrida e pulos. Dificuldades como oclusões parciais, mudanças de

ponto de vista, variação na iluminação e ruídos torna necessário o desenvolvimento de algoritmos robustos para o reconhecimento de objetos em imagens. Segundo (KURIAN, 2011), para um reconhecimento mais preciso, os algoritmos necessitam de treinamento, utilizando-se imagens corretas do objeto a ser identificado. Um conjunto grande de imagens, que possuam o objeto de interesse, deve ser passado ao algoritmo que irá aprender através do mapeamento das características apresentadas pela imagem, o que deve ser detectado.

Segundo (KURIAN, 2011), utilizar apenas um padrão de identificação nas imagens como, cores, texturas e pontos de interesse, não é suficiente. Dois objetos podem ser, por exemplo, de cores, dimensões e proporções diferentes. O desafio em projetar algoritmos eficientes, está relacionado com dificuldades em dividir uma imagem em seus objetos constitutivos, em modelar formas genericamente e em abstrair reflexões, texturas, sombras e distorções causadas pelo processo de captura de imagens. Através de treinamentos de formas previamente treinadas podem ser obtidos reconhecimentos mais precisos. Através de treinamentos usando um banco de imagens, formas de possíveis pessoas possam ser identificados com maior precisão. Neste modelo de reconhecimento, enquadram-se as técnicas de Características de Haar, Histograma de Gradientes Orientados e a Aprendizagem de Máquina.

2.2.1.1 Características de Haar

O conceito foi proposto por (VIOLA; JONES, 2001) baseado no trabalho do matemático Alfred Haar de 1909 (HAAR, 1910). A Transformada de Haar consiste na diferença de intensidade entre regiões retangulares na mesma imagem buscando identificar determinados padrões. O método é sustentado por três pilares: treinamento de classificadores, classificadores em cascata e a imagem integral conforme apresentado na Figura 2.1.

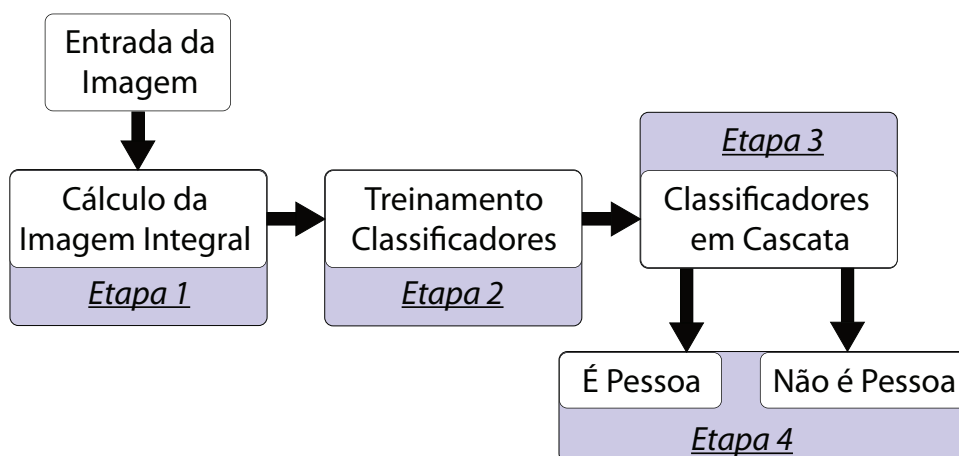


Figura 2.1 – Processo de Características Haar, (Fonte: Adaptado de (VIOLA; JONES, 2001)).

Na etapa 1, conforme a Figura 2.1, é realizado o cálculo da imagem integral, formado pela soma dos pixels acima e à esquerda de determinado ponto. A partir da imagem integral, conjuntos de características podem ser adquiridos com o produto da diferença entre a soma dos pixels de regiões retangulares. Estas características são determinadas pela subtração da média dos valores dos pixels da região considerada escura com a região clara. Caso a diferença estiver acima de um limiar, ou seja, realmente existe uma região clara e outra escura, então a característica existe. Para o cálculo da imagem integral, considere um ponto (x,y) qualquer na imagem, o valor é o resultado do somatório da intensidade dos níveis de cinza de todos os pixels acima e à esquerda de (x,y) , incluindo o próprio ponto. Com o valor da integral calculado, um padrão pode ser identificado através de *Haar-like*. *Haar Likes* são máscaras retangulares resultante da subtração dos valores da intensidade dos pixels de uma determinada região com uma região vizinha (VIOLA; JONES, 2001).

Na segunda etapa, realiza-se o treinamento dos classificadores. Para isso são coletadas imagens positivas e negativas com base no cenário alvo de reconhecimento. As imagens positivas são partes de uma imagem completa que possuem as características ou o objeto que deve ser reconhecido pelo sistema. As imagens negativas são formadas por características e/ou objetos que não devem ser reconhecidos. O classificador é responsável por analisar as imagens positivas e negativas e definir quais as características serão utilizadas para identificar determinado objeto em determinada imagem (VIOLA; JONES, 2001).

Para o treinamento, utiliza-se um método chamado *Boosting* (SCHAPIRE, 1990), que gera um classificador de alta precisão pela combinação de classificadores de média precisão. Os classificadores de média precisão são combinados corrigindo os erros uns dos outros, aumentando assim sua precisão. Os classificadores percorrem as imagens positivas e negativas, identificam as características e armazenam em um arquivo *Extensible Markup Language (XML)* um conjunto de árvores de decisão que usam características *haar-like* (simetrias) para fazer a classificação.

Na etapa 3, os classificadores em cascata são executados na imagem, percorrem a mesma e aplicam a classificação realizada anteriormente em busca de características semelhantes às pretendidas. As características são formadas por retângulos claros e escuros. A cada região analisada, é realizada a subtração da imagem integral da região clara pela imagem integral da região escura. Se o resultado for maior que um determinado limiar significa que a característica existe. Dessa forma na última etapa o algoritmo retorna se é ou não uma pessoa.

2.2.1.2 Histogramas de Gradientes Orientados

Uma imagem possui um conjunto de características (formas, bordas, cores), que dão forma a objetos. O agrupamento dessas características em um vetor escalar é denominado descritor de imagens. Cada objeto é representado por um ponto no espaço R^n , no qual n representa o número de características. O algoritmo de Histogramas de Gradientes Orientados (HOG) (DALAL; TRIGGS, 2005), é um método baseado na avaliação de orientações de histogramas de gradientes locais normalizados de imagem. Os histogramas de gradientes são calculados em uma grade de células igualmente espaçadas na imagem. Ele parte do princípio que um objeto com suas formas pode ser descrito muitas vezes pela intensidade dos gradientes ou a direção das bordas. Com isso, o algoritmo calcula o histograma da orientação dos gradientes e retorna como resultado um vetor de histogramas. Através de classificadores, os histogramas obtidos podem identificar formas, como círculos, retângulos ou qualquer outra forma que permita o treinamento.

O algoritmo de HOG é composto por 6 etapas entre a entrada da imagem e o retorno do resultado final, conforme pode ser visto na Figura 2.2.

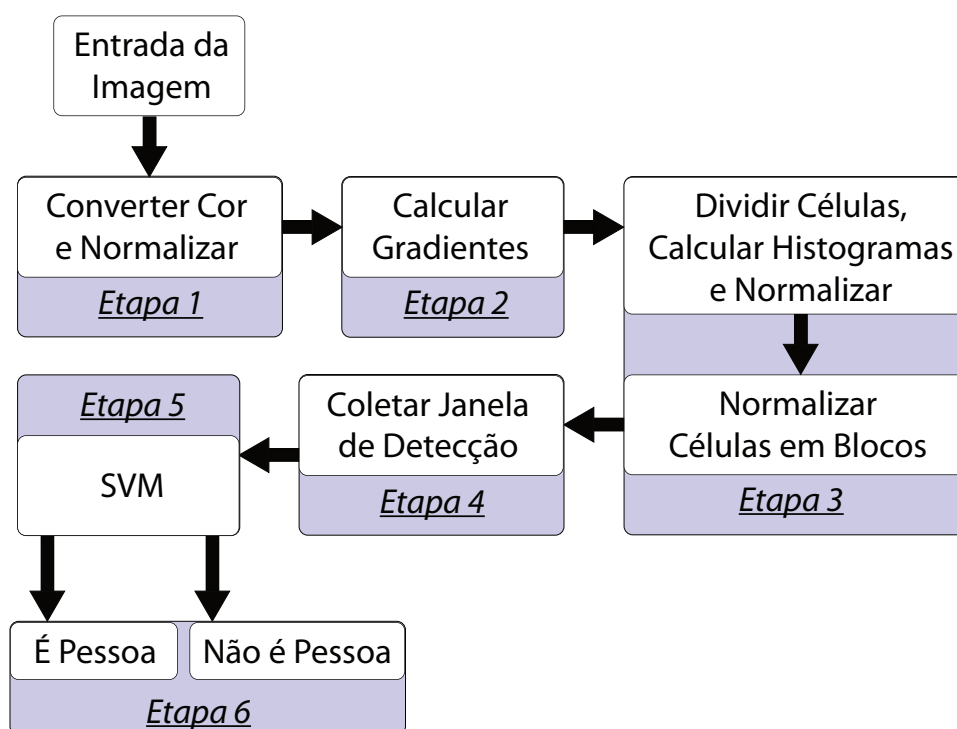


Figura 2.2 – Processo de HOG (Adaptado (DALAL; TRIGGS, 2005)).

A primeira etapa do mecanismo de funcionamento de HOG descrito na Figura 2.2 consiste em converter a imagem para tons de cinza e normalizar a iluminação da imagem de forma diminuir o ruído. Nessa etapa ocorre um aumento no contraste das imagens em tons de cinza, e busca-se ajustar a iluminação das imagens para

se obter um ganho de desempenho. Após, na etapa 2, são calculados os gradientes visando a detecção de bordas. Conforme Dalal e Triggs (DALAL; TRIGGS, 2005), o gradiente pode ser obtido através da derivada de cada pixel ao subtrair o pixel superior do inferior e também o pixel da direita ao da esquerda. Deste modo, pode-se identificar a magnitude de cada pixel e consequentemente a sua orientação G_x e G_y . G_x e G_y correspondem as coordenadas que sucedem determinado pixel em direção a uma possível borda na imagem. O próximo passo, na etapa 3, é dividir a imagem em pequenas partes, que são chamadas de células. Para cada célula é computado um histograma próprio e normalizado. Algumas células são agrupadas em uma região maior denominada bloco, no qual o valor do histograma das células é acumulado e utilizado para normalizar todo o bloco. Os blocos servem para mapear a imagem, e através deles normalizar e varrer toda a imagem a procura de objetos a serem identificados. A normalização dos blocos é feita de forma a sobrepor 50% de outro bloco e cada bloco terá sua normalização calculada separadamente (DALAL; TRIGGS, 2005).

Após todos os blocos serem obtidos, uma janela de detecção HOG é construída na etapa 4. A janela HOG é gerada a partir da junção de todos os blocos, células e histogramas. O detector de janela possibilita a varredura da imagem que ocorre da esquerda para a direita e da parte superior para parte inferior da imagem. Isso acontece para encontrar os objetos pretendidos que possam estar presentes. Por fim, na etapa 5, define-se a forma gerada e retorna-se a descrição da imagem. A decisão sobre definir se a forma gerada é o objeto pretendido, fica a cargo de uma Máquinas de Vetores de Suporte (SVM). E então como resultado na etapa 6 retorna, se a imagem é composta por uma pessoa ou não.

2.2.1.3 Classificação

Uma imagem é composta por um agrupamento de objetos que podem variar em cores, formas e aparecer em diferentes perspectivas. O primeiro passo para um programa de computador compreender semanticamente uma imagem, é extrair suas características de forma eficiente e eficaz (TIAN et al., 2013). A extração das características de uma imagem é realizada através de algoritmos como Haar e HOG, vistos nas subseções acima. O agrupamento de características determina padrões na imagem que podem caracterizar ou não objetos. Para definir se os padrões apresentados na imagem são ou não objetos, um algoritmo de aprendizado deve ser executado para aprender e posteriormente classificar (TIAN et al., 2013).

O aprendizado de máquina realiza um raciocínio indutivo, ou seja, induz conclusões a partir de exemplos reais do objeto pretendido, que são fornecidos ao algoritmo (MICHALSKI; CARBONELL; MITCHELL, 2013). A classificação é uma parte do aprendizado de máquina. O objetivo principal da classificação é aprender a classificar

objetos desconhecidos, porém que possuam características semelhantes àsquelas dos objetos passados por aprendizado. Um classificador deve possuir a habilidade de analisar uma imagem e com base no aprendizado, concluir se aquele objeto identificado é ou não o objeto pretendido, ou ainda em caso de haver mais de uma classe, apontar para qual classe pertence.

Os algoritmos de aprendizado são divididos em 2 (dois) grupos, supervisionados e não supervisionados (MONARD; BARANAUSKAS, 2003). A aprendizagem supervisionada é utilizada quando existe uma base de exemplos grande, em que as classes de cada instância são conhecidas. Já a aprendizagem não supervisionada, não possui classes bem definidas e baseia-se no agrupamento de características. Neste tipo de aprendizagem, o algoritmo deve descobrir sozinho os padrões e categorias.

No escopo deste trabalho de mestrado, as características buscadas são conhecidas, portanto adotou-se a aprendizagem de máquina supervisionada. Os algoritmos utilizados no desenvolvimento da abordagem proposta (descrita no Capítulo 4) são:

- Adaboost (do inglês, *Adaptive Boosting*): é um dos mais conceituados algoritmos de boosting existentes, (FREUND; SCHAPIRE, 1997) e sua classificação é baseada na forma dos objetos a serem classificados. *boosting* é um algoritmo de aprendizado de máquina que combina diversos classificadores fracos (algoritmos de baixa acurácia, um conjunto de funções simples de classificação) para ao final do processo obter uma classificação forte (SCHAPIRE, 1990). A função do AdaBoost é identificar um conjunto de características que formarão o classificador forte de modo a permitir que um conjunto de entrada possa obter uma classificação melhor. Esse conjunto de entrada no caso do Haar Likes são as imagens positivas e negativas. O treinamento com Adaboost é formado por múltiplas iterações de classificação utilizando classificadores de baixa acurácia. A cada iteração os classificadores vão identificando os erros e corrigindo-os. Ao final do processo pode-se obter uma acurácia maior.
- Máquinas de Vetores de Suporte (SVM): são algoritmos de aprendizado introduzidos por Cortes e Vapnik (CORTES; VAPNIK, 1995). A ideia principal é garantir uma alta capacidade de generalização através do mapeamento de entradas. Foi projetado para problemas binários com extensão para problemas multi-classe (BEGG DANIEL T.H. LAI, 2007). O classificador é definido por um hiperplano de separação de objetos gerado pelo algoritmo que categoriza novos exemplos (CRISTIANINI; SHAW-TAYLOR, 2000). O SVM realiza a separação de um conjunto de objetos com diferentes classes, baseado em características pré-

definidas e conhecidas. Para fins de exemplificação, supomos que temos bolas vermelhas à esquerda e azuis à direita. O classificador então traça uma linha, chamada de hiperplano, que separa as cores em classes. Dessa forma, uma nova bola que entrar na imagem à esquerda do hiperplano, será categorizada como vermelha.

2.2.1.4 *Deep Learning*

Redes Neurais consistem em múltiplos processadores conectados entre si, chamados neurônios. Uma rede neural pode ser composta por diversas camadas de neurônios, a qual é denominada *Deep Learning*. *Deep Learning* permite que modelos computacionais compostos de múltiplas camadas de processamento aprendam representações de dados com múltiplos níveis de abstração. O aprendizado profundo descobre uma estrutura complexa em grandes conjuntos de dados usando o algoritmo de retro-propagação para indicar como uma máquina deve alterar seus parâmetros internos que são usados para calcular a representação em cada camada da representação na camada anterior (SCHMIDHUBER, 2015).

Dentre as técnicas de *Deep Learning* que vem ganhando destaque na detecção de pessoas, cita-se o YOLOV3 (REDMON et al., 2016). O YOLOV3 é uma rede neural convolucional de ponta a ponta que detecta objetos com base em predições de caixas delimitadoras e probabilidades de classe. O algoritmo You Only Look Once (YOLO) proposto por Redmon (REDMON et al., 2016), classifica e localiza o objeto em apenas uma etapa e obtém a posição e a categoria do objeto diretamente na camada de saída. No YOLO, a detecção de objetos é tratada como um problema de regressão, o que melhora a velocidade da operação atendendo aos requisitos em tempo real. O YOLOV3 usa os recursos de adaptação da rede com base no Darknet-53, utilizando 53 camadas convolucionais. Tomando por base as potencialidades associadas ao YOLOV3 apresentadas na literatura, escolheu-se avaliar o desempenho da técnica de *Deep Learning* YOLOV3 na contagem de passageiros no cenário alvo (ver Capítulo 5).

2.3 CONCLUSÕES DO CAPÍTULO

Neste capítulo, apresentou-se os fundamentos teóricos para o desenvolvimento de um sistema para a detecção e contagem de pessoas no cenário alvo (descrito no Capítulo 4). Primeiramente, foram apresentados os conceitos de Cidades Inteligentes, Mobilidade Inteligente e Internet das Coisas. Protocolos de comunicação para Internet das Coisas como, CoAP e MQTT também foram analisados.

Na Seção 2.2 foi abordada a Visão Computacional e o seu potencial para

contagem dos passageiros. Foram apresentadas as principais técnicas para detecção de objetos que podem ser aplicadas no desenvolvimento da contagem das pessoas que embarcam e desembarcam do ônibus. Dentre as técnicas discutidas, duas foram destacadas, *Haar Like* e HOG por terem uma grande utilização na literatura, conforme apresentado no Capítulo 3.

No capítulo a seguir, são apresentados os trabalhos relacionados com a proposta deste trabalho, detecção de contagem de passageiros em transporte público urbano e coletivo.

3 TRABALHOS RELACIONADOS

Neste capítulo, apresentam-se alguns trabalhos que lidam com o tema da contagem de pessoas através de câmeras com foco em transporte urbano de passageiros. Na Seção 3.1, é descrita uma classificação das abordagens investigadas, em relação a um conjunto de propriedades elencadas como essenciais para a detecção e contagem de pessoas em transporte urbano.

3.1 CONTAGEM DE PASSAGEIROS EM TRANSPORTE URBANO

O grande número de pessoas trafegando pelas estações de transporte, dificulta a obtenção de uma contagem precisa sobre o número de passageiros. A Visão Computacional permite coletar imagens das estações e da entrada/saída dos ônibus, analisá-las e efetuar a contagem de pessoas.

As abordagens encontradas na literatura que tratam da contagem de passageiros por meio de câmeras podem ser classificadas de acordo com as seguintes propriedades:

- **Propriedades da Câmera:** Existem duas propriedades importantes na contagem, Posição e Profundidade. A Posição, interfere no formato da pessoa e as câmeras podem possuir diferentes posições em um ônibus (p.ex. Zenital, Lateral e Frontal). A Profundidade, interfere no tamanho que a pessoa representa na imagem. Algumas câmeras possuem um sensor de profundidade que facilita na detecção;
- **Tamanho da Pessoa na Imagem:** Relacionada ao espaço que a pessoa ocupa na imagem. Quanto mais próxima a pessoa está da câmera, maior a dificuldade de identificação;
- **Detecção e Rastreamento:** A detecção, refere-se aos algoritmos utilizados para identificar as pessoas na imagem. O rastreamento, refere-se aos algoritmos utilizados para realizar o acompanhamento do objeto identificado e para posterior contagem;
- **Processamento e Agregação:** Os vídeos do embarque e desembarque de pessoas necessitam ser processados para que a contagem possa ser realizada. Além disso, os dados precisam ser disponibilizados. O processamento dos vídeos pode ser embarcado no ônibus através de dispositivos especiais, por exemplo, microcontroladores ou enviados via rede para fora do ônibus e posterior processamento.

Na Figura 3.1, apresenta-se uma taxonomia com base nas propriedades apresentadas anteriormente.

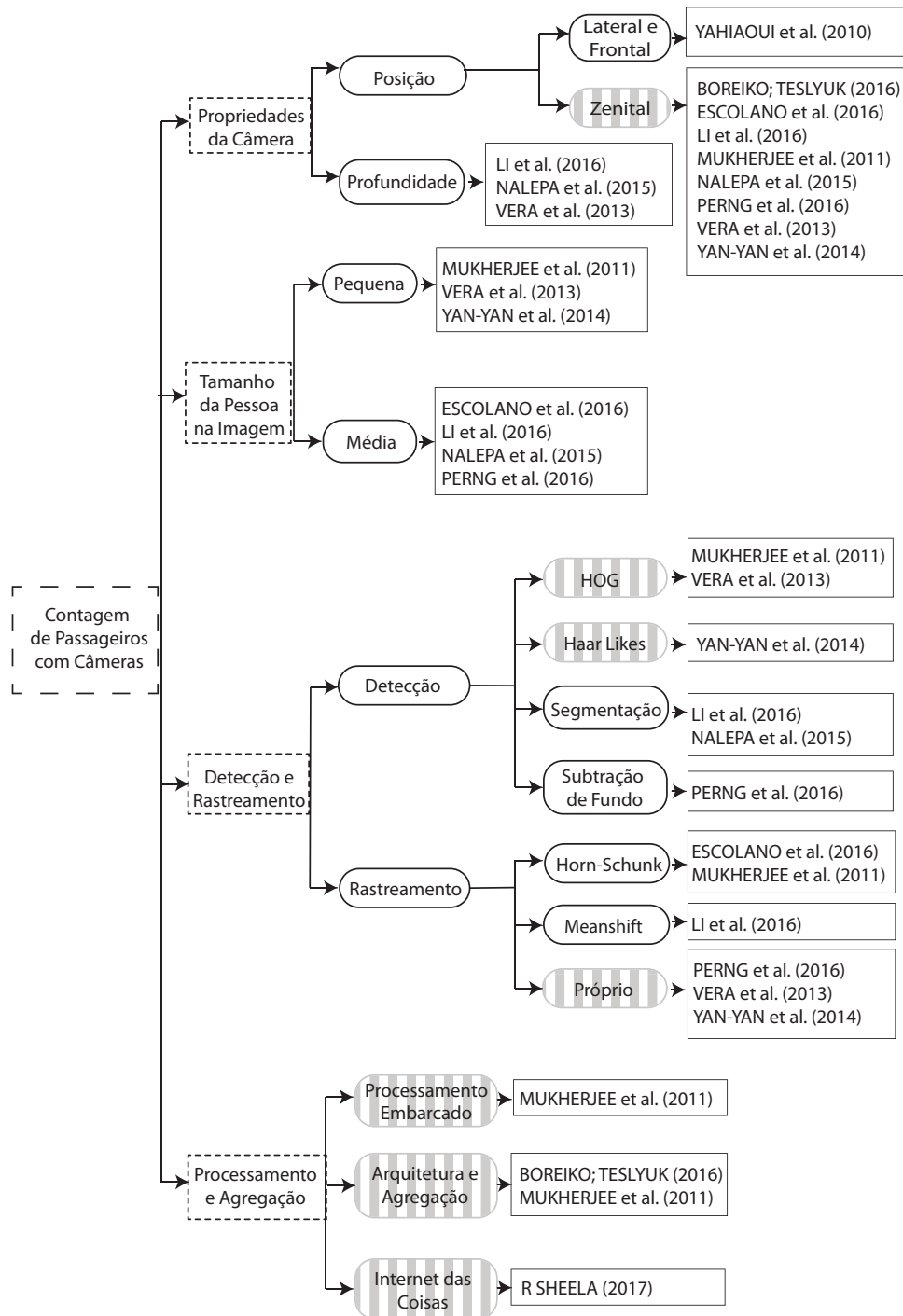


Figura 3.1 – Taxonomia de Sistemas para Contagem de Passageiros, Fonte: Próprio Autor.

O trabalho aqui apresentado, com base na taxonomia acima descrita, utilizou como base algumas propriedades, que podem ser identificadas na Figura 3.1 através das hachuras. A posição das câmeras utilizada nas coletas, foi a zenital, pois na literatura apresentou os melhores resultados com menor custo computacional. Devido

a estrutura do ônibus ser baixa comparado a alguns ônibus da literatura e as câmeras utilizadas não possuem sensor de profundidade, as pessoas detectadas ficavam muito próximas à câmera. Dessa forma o tamanho das pessoas na imagem apresenta uma característica diferente das previstas na literatura, ocupando um grande espaço.

Dentre os algoritmos de detecção e rastreamento optou-se por aqueles que apresentaram na literatura maior número de utilização com melhores acurácias. Portanto para detecção utilizou-se HOG e Haar Likes e para rastreamento foi desenvolvido um próprio.

Um fator de muita importância para a aplicação deste trabalho é a possibilidade de funcionamento transparente e de forma rápida. A utilização dos conceitos de Internet das Coisas nesse aspectos são importantíssimas. Para isso foram implementados algumas propriedades que poucos trabalhos da literatura utilizaram. O processamento dos dados embarcado, facilita que os dados possam ser transmitidos já tratados e otimizados, o que economiza e facilita na transmissão para fora da rede interna do ônibus. Uma arquitetura possibilita que exista uma estruturação desde a coleta dos vídeos, tratamento até a utilização final.

Na próxima seção serão apresentados detalhes das propostas para contagem de passageiros com base na taxonomia descrita na Figura 3.1.

3.1.1 Abordagem para Contagem de Passageiros com Câmeras

Alguns trabalhos da literatura abordam a contagem de pessoas em ônibus através de visão estéreo, como (YAHIAOUI; KHOUDOUR; MEURIE, 2010). O autor utiliza duas câmeras para a captura de imagens. Uma câmera coleta as imagens de cima e outra lateralmente. Um pareamento das câmeras é realizado, ligando os eixos que permitem criar um enlace entre as duas posições. As técnicas, buscam através da similaridade nos tons de cinza entre as duas imagens, eliminar o fundo, identificar pixels vizinhos parecidos e detectar bordas. Deste modo, consegue-se identificar as pessoas entrando/saindo na imagem e acompanhar seu fluxo, possibilitando efetuar a contagem das mesmas. O processamento dos vídeos foi realizado externamente ao ônibus, assim como o processamento da contagem.

(MUKHERJEE et al., 2011), apresenta um *framework* para detecção e rastreamento de pessoas em estações de trem. O *framework* é dividido em 3 (três) partes: detecção das pessoas, rastreamento e validação. A etapa de detecção é realizada através de câmeras colocadas em cima das entradas da estação. Utilizando o algoritmo de *Hough* (YUEN et al., 1990) para detectar formas circulares e também o algoritmo de *HOG* (DALAL; TRIGGS, 2005), o sistema identifica pessoas pelas formas circulares de suas cabeças. Posteriormente, realiza-se a etapa de rastreamento. O sis-

tema analisa os quadros posteriores de movimento através da identificação do fluxo óptico pelo método *Horn-Schunk* (HORN; SCHUNCK, 1981). O passageiro é acompanhado até sair do campo de visão da câmera. Para completar, o sistema executa a etapa de validação a fim de evitar falsos positivos na identificação de pessoas, como outras partes do corpo (p.ex. mãos), ou até duplicações. Com base em espaço temporal é realizado uma análise para identificar se a forma encontrada segue um padrão de movimento, caso não siga, a forma é descartada. Os autores não especificam se a contagem é interna ou externa ao trem. Também não mencionam como a transmissão dos dados da contagem é realizada.

No trabalho de (YAN-YAN et al., 2014), os autores propõe um sistema de contagem de pedestres em uma estação de metrô com base na detecção de *Haar Likes* (VIOLA; JONES, 2001). As imagens são coletadas através de uma câmera alocada na posição zenital e processadas em um servidor. O sistema é dividido em duas etapas, treinamento e detecção. O treinamento ocorre com um determinado número de amostras através do algoritmo *Adaboost* (FREUND; SCHAPIRE, 1997). Aplica-se o treinamento nas imagens para identificação das cabeças de pessoas, e através de um cálculo de posicionamento é feito o rastreamento da pessoa no alcance da câmera.

No trabalho de (NALEPA; SZYMANEK; KAWULOK, 2015), propõe-se um algoritmo para contar pessoas a partir de imagens capturadas por câmeras com profundidade. Uma câmera do tipo Kinect foi instalada dentro do ônibus, sobre a porta em posição zenital. A câmera Kinect possui um sensor que permite identificar a profundidade dentro da imagem. Através do algoritmo, executam-se operadores morfológicos seguidos por uma segmentação de profundidade da imagem e um agrupamento de segmentos. Os processamentos foram realizados externamente, por um computador dotado de um processador i7 com 16 GB de RAM.

(BOREIKO; TESLYUK, 2016) apresentam um modelo estrutural para contagem de passageiros e rastreamento em transporte público. O sistema realiza a contagem através de duas câmeras alocadas no teto e acima das portas do ônibus para identificar as pessoas que passam. Durante o percurso, a cada parada, quando as portas do ônibus se abrem o sistema grava as imagens. Ao fechar as portas, um controlador Raspberry Pi agrega as imagens gravadas juntamente com as informações de localização geográfica coletadas através de um Sistema de Posicionamento Global (GPS). O microcontrolador envia os dados para um servidor externo por um modem 3G. O servidor calcula o número de passageiros e a posição do veículo pelas coordenadas. Por fim, as informações coletadas e processadas são apresentadas através de uma representação por Redes de Petri. Neste trabalho, observa-se que todo o processamento é realizado externamente ao ônibus podendo agregar um alto custo na transmissão dos vídeos pela rede 3G. O foco do trabalho está na estrutura

de contagem e na apresentação da rede de transporte.

No projeto de (ESCOLANO et al., 2016) na cidade de Manilla, nas Filipinas, câmeras são instaladas nas portas dos ônibus para coletar as imagens. O sistema, baseia-se em uma estimativa de fluxo óptico para detectar e acompanhar o embarque e desembarque dos passageiros. O fluxo óptico utiliza sequências de imagens para identificar o sentido em que o passageiro passa. Para o cálculo da detecção nas imagens, a função *Horn-Schunk* (HORN; SCHUNCK, 1981) é aplicada. A função recebe uma imagem anterior e uma imagem atual como entrada retornando como resultado, o fluxo. Esses dados são transmitidos para uma central de despachos, onde um controlador de Lógica Fuzzy determina o horário do envio de um novo ônibus.

No trabalho de (LI et al., 2016), utiliza-se uma câmera kinect da Microsoft na posição zenital nas portas do ônibus para realizar as gravações. A câmera tem a capacidade de obter uma imagem de profundidade, combinada com uma imagem Red, Green e Blue (RGB). Neste trabalho são realizados processamentos morfológicos para eliminar ruídos. Aplica-se a segmentação de imagem através de métodos gaussianos para fazer a subtração de fundo. Para detectar as pessoas é realizado um aprendizado de máquina, utilizando-se SVM para treinar o sistema para identificar as cabeças das pessoas. O rastreamento das pessoas é realizado através da técnica de *meanshift* que possibilita realizar uma predição do caminho de uma pessoa. O sistema possui duas linhas limítrofes, nas quais uma pessoa deve cruzar para ser contabilizada. Os autores não detalham os mecanismos utilizados para realizar a comunicação, ou os equipamentos adotados para o processamento.

No trabalho de (PERNG et al., 2016), um algoritmo composto por três etapas foi apresentado: Subtração de fundo, Detecção de pessoas e Rastreamento e Contagem. Com base em uma entrada de vídeo, o algoritmo aplica uma técnica de subtração de fundo para detectar movimentos na imagem. Através de processamentos morfológicos, busca-se eliminar ruídos e masterizar a imagem. Com a imagem obtida, o algoritmo converte a imagem resultante em uma imagem digital binária e separa os blocos de pixels agrupados. O algoritmo realiza um escaneamento na imagem, visando identificar características, como cabeça e ombro. Quando a pessoa é identificada, o sistema faz um rastreamento da pessoa pela imagem. Se a pessoa ultrapassa uma linha de contagem pré-definida, a pessoa é contabilizada. Neste trabalho, não são detalhados os mecanismos de comunicação ou processamento das imagens.

Com base na análise da literatura, observou-se que normalmente são utilizadas duas posições de câmera, zenital e stereo. Na posição em stereo utiliza-se duas câmeras, uma com vista de cima e outra com vista lateral. Na posição zenital a câmera é alocada sobre a cabeça voltada em visão reta para o chão. Algumas câmeras na posição zenital, são do tipo Kinect. Uma câmera do tipo kinect possui um sensor

que apresenta nas imagens coletadas uma variável de profundidade que pode facilitar nas segmentações. Outro aspecto que contribui para a segmentação é análise do tamanho da pessoa em relação à imagem.

O trabalho aqui abordado, utilizará câmeras na posição zenital alocada na porta do ônibus para realizar a contagem dos passageiros porém sem sensor de profundidade pelo alto custo de implementação. Dessa forma as imagens, coletadas no ônibus serão próximas às cabeças e assim as pessoas ocuparão um espaço grande na tela. Para detecção, serão abordados os algoritmos de Haar e HOG, em vista dos resultados apresentados pela literatura e para o rastreamento será desenvolvido um algoritmo próprio. O trabalho também irá abordar características que grande parte dos outros trabalhos encontrados não apresentou. O sistema será estruturado por uma arquitetura lógica e de implementação e proverá o processamento embarcado no ônibus, não necessitando assim enviar os vídeos inteiros para fora do ônibus. Dessa forma, um sistema de agregação também será implementado baseado no conceito de Internet das Coisas para prover integração e comunicação dos dados de forma transparente.

3.2 CONCLUSÕES DO CAPÍTULO

Neste capítulo foi realizado um levantamento bibliográfico sobre contagem de passageiros em ônibus através da Visão Computacional.

Os mecanismos para a contagem identificados na literatura se dividem em duas categorias: detecção e rastreamento. Os algoritmos de HOG e Haar Likes são os mais empregados na detecção. Os algoritmos de rastreamento na maior parte dos trabalhos são de autoria própria, com base em cálculos da sequência de pixels.

Nas imagens coletadas para os experimentos realizados neste trabalho (ver Capítulo 5), constatou-se que as pessoas ocupam um espaço grande na imagem, quando as câmeras são colocadas em posição zenital. Se a câmera for alocada lateralmente ou de frente para as pessoas que entram, o tamanho das pessoas fica menor em relação a imagem, porém a oclusão de pessoas entrando ou saindo é maior e dificulta a contagem. Para corrigir o problema de oclusão e altura, o trabalho de (YAHIAOUI; KHOUDOUR; MEURIE, 2010), usa duas câmeras na posição stereo. Entretanto, a utilização de duas câmeras pode demandar um processamento maior do microcontrolador.

A proposta deste trabalho de mestrado é uma abordagem para detecção e contagem de pessoas em ônibus urbano. Para atingir este objetivo, câmeras serão posicionadas nas portas do ônibus. Microcontroladores serão instalados no ônibus para realizar o processamento embarcado das imagens utilizadas na contagem.

No Capítulo 4, será apresentado com maior detalhes a arquitetura da abordagem proposta para a detecção e contagem de pessoas.

4 ABORDAGEM DESENVOLVIDA

Neste capítulo, apresenta-se a abordagem proposta para a contagem de passageiros com base em visão computacional e utilizando-se os conceitos de Internet das Coisas. O sistema realiza a contagem automática de pessoas que passam pela porta do ônibus através de técnicas de detecção de objetos e análise de imagens obtidas por câmeras de vídeo. A arquitetura do sistema para detecção e contagem de pessoas foi especificada com a finalidade de oferecer suporte para a contagem do fluxo de passageiros (entrada e saída) em cada trajeto percorrido pelo ônibus e prover informações coletadas no ônibus para a empresa de transporte coletivo. A contagem, detecção de pessoas e a transmissão de dados é realizada através de equipamentos de baixo custo embarcados no ônibus.

4.1 ARQUITETURA DO SISTEMA

A Figura 4.1, representa a arquitetura lógica proposta para o sistema de contagem de passageiros, contendo as seguintes camadas: Percepção, Tratamento, Interface de Comunicação e Dados e Aplicação.

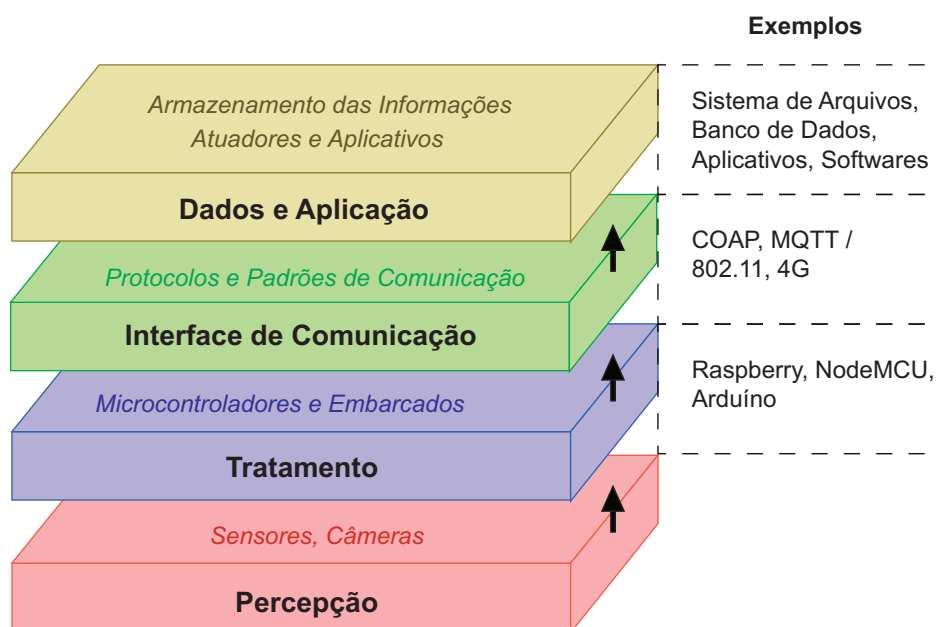


Figura 4.1 – Arquitetura Lógica do Sistema, Fonte: Próprio Autor.

A Camada de Percepção tem a função de perceber o ambiente e coletar informações apresentadas por ele através de periféricos, como sensores (p.ex. temperatura, umidade, presença, infravermelho) e câmeras. Na abordagem proposta, utilizou-se câmeras para realizar a gravação do fluxo de entrada e saída do ônibus. Após os

dados serem coletados, necessita-se tratar estes dados, uma vez que a informação obtida pode ser de uma pessoa ou várias, ou mesmo um objeto como, uma mala ou uma mochila transportada por um passageiro.

O tratamento das imagens coletadas é realizado pela Camada de Tratamento, que contém algoritmos cuja a função é identificar pessoas nas imagens e contabilizá-las à medida que passam pela porta do ônibus (Entrada ou Saída). No escopo deste trabalho, a Camada de Tratamento é composta por placas de desenvolvimento que tem capacidade de processamento, como Arduíno, NodeMCU ESP32 e Raspberry Pi.

Na operacionalização do sistema, as câmeras estão interligadas a um Raspberry Pi 3B, que faz o processamento da Camada de Tratamento. As imagens são obtidas e armazenadas pela Camada de Percepção através de câmeras e devem ser processadas pela Camada de Tratamento. Os dados processados são transmitidos pela Camada de Interface de Comunicação.

Conforme descrito na Figura 1.1 da Seção 1.1, um ônibus pode conter duas ou três portas. Dessa forma, cada porta deverá realizar a sua contagem independente. Neste contexto, existe a necessidade de haver comunicação entre a contagem das portas para que o sistema possa agregar os dados de cada porta e realizar uma contagem efetiva do número de passageiros que entraram ou saíram pelas portas em cada uma das paradas realizadas pelo ônibus.

A comunicação entre as portas, bem como outros aspectos relacionados a comunicação do sistema com a rede externa ao ônibus para disponibilização dos dados são gerenciadas pela Camada de Interface de Comunicação (Figura 4.1). No sistema, os dados gerados no ônibus sobre a contagem podem ser enviados para a empresa através de uma tecnologia de comunicação, por exemplo, redes celulares (GPRS, EDGE, 3G ou 4G), ou descarregados nos terminais ou garagem da empresa por meio de uma rede sem fio do tipo IEEE 802.11 b/g/n.

A Camada de Dados e Aplicações concentra a função de armazenamento das informações tratadas e apresentação dos dados aos usuários finais. A implementação desta camada pode estar em uma nuvem computacional ou em um servidor externo da empresa.

O sistema ainda é composto por uma arquitetura de implementação, composto pelo hardware responsável por executar a arquitetura lógica, conforme apresentado na Figura 4.2.

A arquitetura de implementação é composta pelos módulos: (I) Detecção e acionamento de porta (Camada de Percepção da Figura 4.1); (II) Contagem de passageiros (Camada de Tratamento); (III) Agregação e Comunicação (Camada de Tratamento, Camada de Interface de Comunicação); (IV) Aplicação (Camada de Dados e

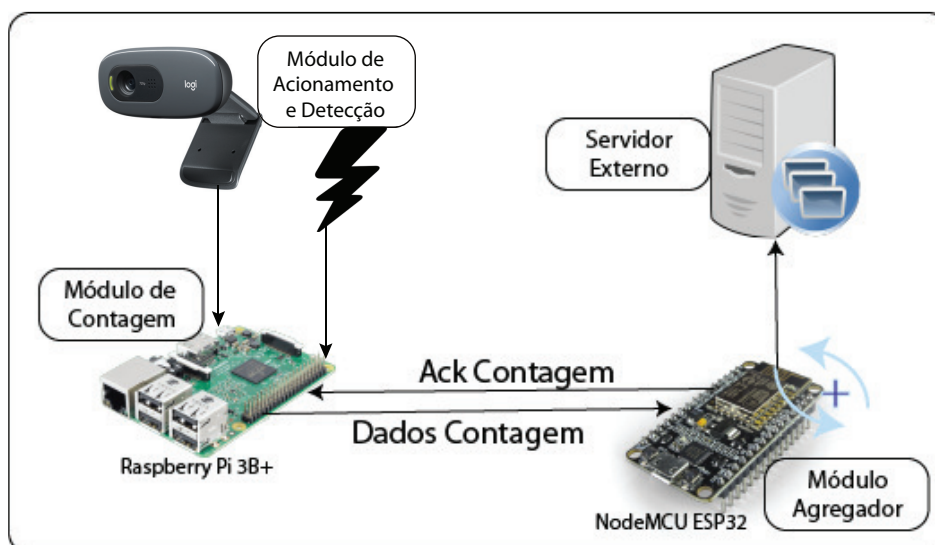


Figura 4.2 – Arquitetura de Implementação do Sistema, Fonte: Próprio Autor.

Aplicação).

O módulo de Detecção e Acionamento de Porta é responsável pelo acionamento de abertura da porta e coleta das informações dos passageiros que entram. O acionamento de porta é realizado pelo sistema interno do ônibus, que envia um pulso analógico para o módulo de Contagem. O pulso emitido possui 24 volts e passa por um conversor de voltagem que reduz a 5 volts antes de chegar ao módulo de Contagem. A detecção das pessoas que entram é realizado por equipamentos de coleta, como câmeras ou sensores. O módulo de Detecção e Acionamento de Porta opera na camada de Percepção na Arquitetura Lógica.

O módulo de Contagem é responsável por executar os algoritmos que irão tratar os dados coletados pelo módulo de Detecção. O módulo é formado por microcontroladores que possuem poder de processamento como Raspberry Pi e opera na camada de Tratamento. Geralmente um ônibus possui 2 ou mais portas, dessa forma haverá mais de um módulo de contagem em cada ônibus, um a cada porta.

O módulo de Agregação e Comunicação é responsável por agregar os dados obtidos pelo módulo de Contagem de cada porta do ônibus. Ele é formado por microcontroladores com capacidade de processamento e gerenciamento de comunicação sem fio e opera nas camadas de Tratamento e Interface de Comunicação.

O módulo de Aplicação é responsável por armazenar os dados de contagem de cada ônibus e disponibilizar para a empresa e usuários finais. É formado por servidores, banco de dados, aplicativos e softwares e opera na camada de Dados e Aplicação. Esse módulo, vale ressaltar que não foi implementado completamente, pois apenas recebe informações vindas do módulo de Agregação.

Na seção seguinte, apresenta-se o mecanismo para a detecção e contagem de passageiros. Na solução, definiu-se que o sistema entra em operação, quando houver um acionamento de abertura de porta do ônibus.

4.2 ALGORITMO DE CONTAGEM

O esquema de funcionamento da solução para a detecção e contagem de passageiros no ônibus foi organizado nos módulos apresentados na Seção 4.1. O fluxograma do algoritmo de detecção e contagem encontra-se descrito na Figura 4.3.

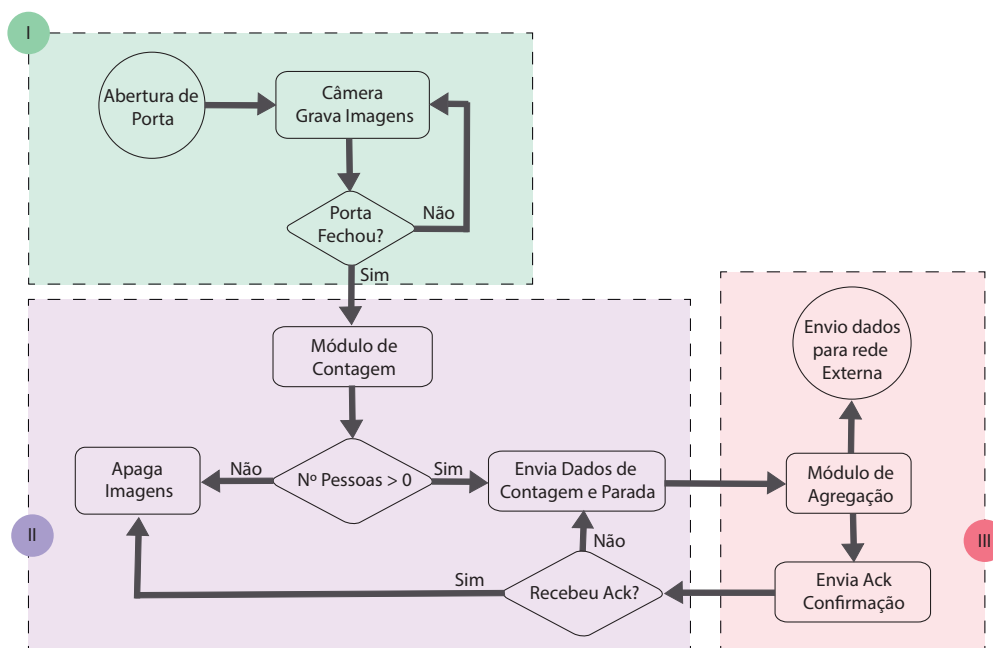


Figura 4.3 – Fluxograma do Algoritmo de Detecção e Contagem, Fonte: Próprio Autor.

O sistema inicia com a abertura de porta do ônibus através do módulo de Detecção e Acionamento na Etapa I conforme a Figura 4.3. Quando o motorista aciona alguma porta, seja para abrir ou fechar, o sistema interno do ônibus emite um pulso elétrico à placa do módulo de Contagem. A recepção do pulso elétrico notifica o sistema que ocorreu um acionamento de porta. No Módulo de Contagem foi adicionada uma câmera para coletar as imagens, juntamente com a placa de desenvolvimento que irá processar as imagens e realizar a contagem, no caso desse projeto, um Raspberry Pi 3B (RASPBERYP.ORG, 2017).

Ao receber o sinal de abertura de porta, o módulo de Contagem de passageiros através do microcontrolador Raspberry Pi recebe as imagens capturadas após a abertura da porta. Esse processo se mantém até a porta ser fechada. Neste momento, um sinal de fechamento de porta é enviado ao Módulo de Contagem que processa e

analisa as imagens para efetuar a contagem de passageiros, então inicia-se a Etapa II da Figura 4.3.

Para fins de contagem, no microcontrolador são armazenados os seguintes dados: número de entradas, número de saídas, identificador do módulo da porta (cada porta do ônibus tem um módulo de Contagem) e identificador de parada. As variáveis, número de entradas, número de saídas, identificador da parada e identificador do módulo da porta tem 3 (três) bytes cada uma, totalizando apenas 12 (doze) bytes no microcontrolador. O identificador de parada é um número inteiro que é incrementado a cada vez que o acionamento de porta é ativado. O identificador de parada, recebe o valor 0 (zero) quando o ônibus termina o trajeto total e retorna para a garagem (local para abastecimento e limpeza) da empresa. O sistema utiliza dados de localização obtidos de um GPS no microcontrolador para identificar o final do percurso.

Conforme mencionado nas seções acima, um ônibus pode conter mais de uma porta, portanto utilizará mais de um módulo de contagem. Consequentemente, faz-se necessário um novo equipamento capaz de agregar as informações coletadas pelas portas, visando prover melhor acurácia e tolerância a falhas na contagem. Denomina-se este equipamento de Módulo Agregador. No escopo deste trabalho, utilizou-se um NodeMCU ESP32. Tanto no módulo de Contagem, quanto no de Agregação deve haver a capacidade de comunicação, que permita transmitir as informações dos módulos de contagem para o agregador, e do agregador para uma rede externa ao ônibus, especificamente para um módulo de Aplicação. O módulo de Aplicação pode ser um servidor em nuvem ou na empresa de ônibus.

Quando a porta fecha e a contagem de passageiros pela análise das imagens termina, pode ocorrer do valor resultante ser igual a zero, por exemplo, se ninguém entrar ou sair do ônibus, a imagem coletada pela câmera não irá detectar pessoas. Se o valor resultante da contagem for igual a zero, o sistema exclui as imagens obtidas. Se a contagem for maior que zero o sistema executa a Etapa III, e o módulo de Contagem nas portas divulga via rede sem fio para o módulo de Agregação o número de passageiros que entraram ou saíram do ônibus entre os eventos de abertura e fechamento de porta. O módulo de Agregação recebe os dados de cada porta, processa e mantém atualizado o número de passageiros no ônibus a cada instante de tempo. Assim que receber a mensagem de atualização do módulo de Contagem, o módulo de Agregação envia uma mensagem de reconhecimento (ACK_Contagem) informando da conclusão do processo. A transmissão de mensagens entre os módulos de Contagem e Agregação é realizada através de uma rede IEEE 802.11 b/g/n com o protocolo de transporte *UDP* e *CoAP* na Camada de Aplicação. Ressalta-se que outras tecnologias de comunicação, podem ser utilizados entre os Módulos de Contagem e Agregação, como Bluetooth Low Energy sobre 6LowPAN, por exemplo.

Em seguida, as imagens armazenadas no módulo de Contagem são eliminadas. Neste ponto do texto, é importante explicar que nos experimentos realizados (ver Capítulo 5), o tamanho dos vídeos gravados entre a abertura e o fechamento da porta têm em média 10 MB podendo alcançar um valor expressivo ao final da rota. Em virtude dessa característica, decorrente da utilização de câmeras, aliada aos problemas provocados pela baixa capacidade de armazenamento e processamento dos microcontroladores tem-se um grande desafio: manter o mínimo de dados nos microcontroladores ou na rede sem prejudicar a contagem de passageiros.

O Algoritmo 4.1 descreve o procedimento para a contagem de passageiros que entram e saem do ônibus. Especificamente, o foco do algoritmo está na identificação dos passageiros por meio das imagens capturadas após o acionamento da abertura da porta. As Figuras 4.4, 4.6 e 4.7 apresentam o fluxo de funcionamento do algoritmo, para facilitar o entendimento.

O Algoritmo representado pelas três figuras acima mencionadas é executado pelo módulo de Contagem e opera na camada de Tratamento. O algoritmo é fundamentado em três funções: identificação, rastreamento e contagem. A identificação é baseada em alguma técnica detecção, por exemplo, Haar Likes (VIOLA; JONES, 2001) ou HOG (DALAL; TRIGGS, 2005). Na técnica de Haar Likes, realiza-se um treinamento com imagens positivas e negativas utilizando um classificador Adaboost. Na técnica de HOG, o treinamento é realizado utilizando um classificador SVM. O classificador após o treinamento gera um arquivo XML com o mapeamento de pixels que contém padrões de pessoas que podem ser identificadas em imagens. Vale ressaltar, que o treinamento que gera o arquivo XML é realizado em um computador externo ao ônibus, com maior poder computacional. Após a identificação da pessoa, utilizando Haar ou HOG, o algoritmo realiza o rastreamento do indivíduo e a contagem ao identificar que uma pessoa atravessou uma linha limítrofe. As técnicas, Haar Likes e HOG, se diferenciam uma da outra na forma como tratam os quadros.

Algorithm 4.1: Contabilização de Passageiros

```

1 positions = [];
2 isCounted = False;
3 classifier = getCascadeClassifier('classificador.xml');
4 SZ_LIMIT1 = 120; /*Tamanho mínimo dos quadros identificados*/;
5 SZ_LIMIT2 = 150; /*Tamanho máximo dos quadros identificados*/;
6 limit_line1 = 50;
7 distance_threshold = 100;
8 in_count = 0;
9 out_count = 0;
10 people_list = [];
11 previous_frame = 0;
12 first_frame = true;
```

```

13 Function findPeople(peoples)
14   for each people ∈ peoples do
15     rectangle_center = [people.getX + people.getWidth/2, people.getY +
16       people.getHeight/2];
17     lowest_distance = 0;
18     closest_index = None;
19     for each p ∈ people_list do
20       distancia = sqrt((p.getX - rectangle_center.getX) ** 2 + (p.getY -
21         rectangle_center.getY) ** 2);
22       if lowest_distance > distancia then
23         lowest_distance = distancia;
24         closest_index = p.index;
25       end
26     end
27     if lowest_distance > 100 then
28       closest_index = None;
29     end
30     if closest_index != None then
31       if lowest_distance < distance_threshold then
32         updatePosition(rectangle_center);
33         count(limit_line1);
34       end
35     else
36       new_person = rectangle_center;
37       people_list.append(new_person);
38     end
39   end
40   if first_frame == true then
41     first_frame = False;
42   end
43 end
44 Function updatePositions(new_position)
45   positions.append(new_position);
46   if lenght(positions) > 100 then
47     positions.pop(0);
48   end
49 end
50 Function onOppositeSides(y_coordinate)
51   val1 = (positions.getY > y_coordinate) and (positions.getY <= y_coordinate);
52   val2 = (positions.getY <= y_coordinate) and (positions.getY > y_coordinate);
53   return val1 || val2;
54 end

```

```

54 Function count(y_coordinate)
55   if onOppositeSides(y_coordinate) then
56     if positions.getY < limit_line1 then
57       if isCounted == False then
58         | in_count=in_count+1;
59       end
60     end
61     else
62       if isCounted == False then
63         | out_count=out_count+1;
64       end
65     end
66   end
67 end
68
69   video = capturaimagem('videodecaptura.avi');
70   nFrames = 0;
71   for todos os frames serem analisados do
72     frame = readNextFrame(video);
73     nFrames += 1;
74     if nFrames < 1 then
75       | previous_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY);
76       | continue;
77     end
78     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY);
79     peoples = classifier.getPeoples(gray, 1, 5, cv2.CASCADE_SCALE_IMAGE,
      (SZ_LIMIT1, SZ_LIMIT1), (SZ_LIMIT2, SZ_LIMIT2));
80     findPeople(peoples);
81   end
82 end

```

No Algoritmo 4.1, inicialmente define-se um vetor que servirá para armazenar as posições das pessoas identificadas (linha 1). Também é definida uma variável de controle que identifica se determinada pessoa já foi contabilizada (linha 2). Um classificador é então definido, com base no treinamento realizado previamente em um computador externo ao ônibus com maior poder computacional. Nas linhas 4 e 5 são definidas algumas constantes responsáveis por estabelecer limites, de tamanho máximo e mínimo para objetos identificados. Esses limites são utilizados para evitar possíveis falsos positivos. Os valores de 120 para o limite mínimo e 150 para o limite máximo foram definidos baseados em uma observação das imagens. Define-se ainda, a linha limítrofe para contagem (linha 6) e distância entre um objeto e outro para fins de rastreamento (linha 7).

Na Figura 4.4, é detalhado o fluxo de processamento do vídeo de cada parada. Um vídeo é formado por uma sequência de quadros e cada quadro contém uma imagem, conforme apresenta a Figura 4.5. O algoritmo recebe o vídeo da parada e processa todos os quadros. Na linha 69 ocorre a captura dos quadros do vídeo e entre as linhas 71 a 81 é realizado o processamento e análise do vídeo, quadro a quadro.

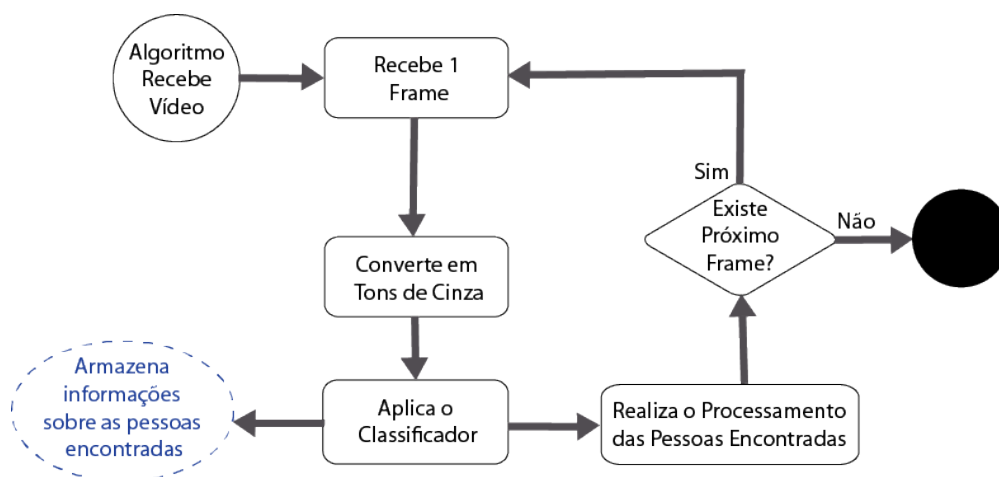


Figura 4.4 – Fluxograma do Processamento do Vídeo no Algoritmo de Contagem, Fonte: Próprio Autor.

No processamento o algoritmo recebe o primeiro quadro (linha 72) e converte em tons de cinza para facilitar a distinção de formas (linha 78). Após, é aplicado ao quadro um classificador (linha 79), que será responsável por identificar as pessoas presentes naquele quadro e armazenar em um vetor. Neste trabalho foram utilizados dois classificadores, Haar Likes e HOG, sendo que o algoritmo utiliza apenas um desses por vez.

Ao identificar as pessoas no quadro, o algoritmo realiza o processamento das mesmas a fim de detectar um embarque ou desembarque e assim efetuar a contagem. Esse processo é demonstrado pela Figura 4.6. Para isso executa-se a função (`findPeople()`) (linha 80). Por fim o algoritmo faz uma verificação nos quadros do vídeo, se houver algum quadro ainda a ser processado, ele o faz, caso contrário o sistema finaliza a contagem.

Com as pessoas identificadas no quadro, conforme apresentado na Figura 4.5, e armazenadas em um vetor local, o sistema realiza o processamento da contagem, como apresentado no fluxograma da Figura 4.6. A função (`findPeople()`) analisa os registros encontrados no quadro e executa uma comparação com a lista de registros anteriores em busca de movimentos. São definidos para isso, os pontos centrais dos retângulos que demarcam as pessoas (linha 15). Com isso, tem-se um vetor local de pessoas identificadas no quadro atual (linha 79) e uma lista global de pessoas identificadas em todo processo de contagem (linha 10).

O algoritmo recebe o vetor de pessoas e realiza o processamento de cada pessoa do vetor. Nesse momento é realizada a comparação da posição atual com todas as pessoas já identificadas pelo sistema para descobrir se a pessoa do quadro já está adicionada ao sistema (linhas 18 a 24). Vale lembrar que o algoritmo guarda em

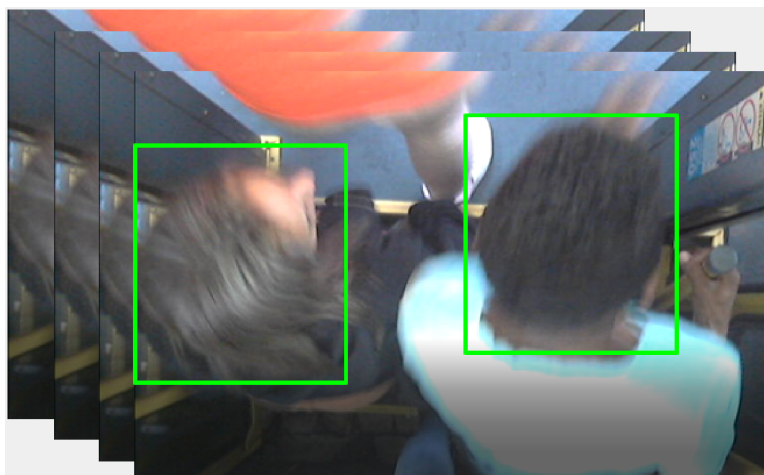


Figura 4.5 – Estrutura de Quadros de um Vídeo e a Detecção de Pessoas, Fonte: Próprio Autor.

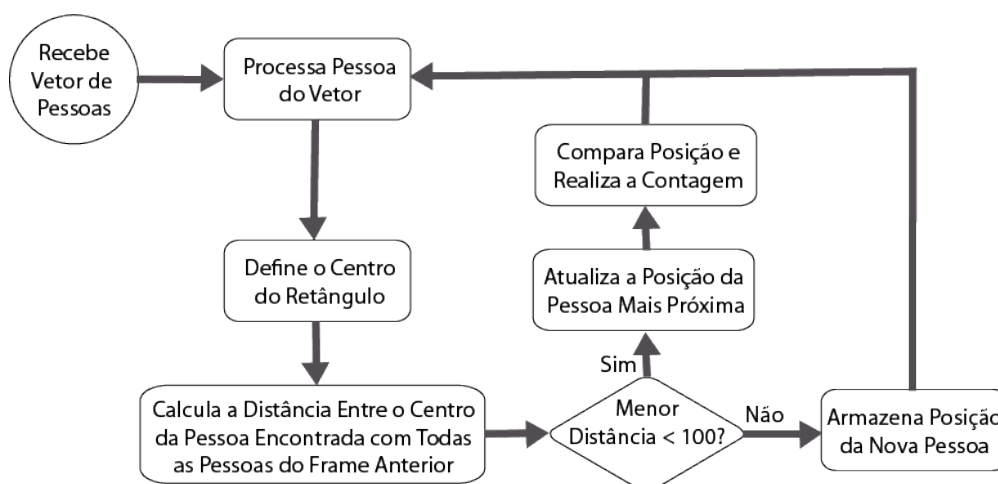


Figura 4.6 – Fluxograma do Processamento de Quadros no Algoritmo de Contagem, Fonte: Próprio Autor.

um vetor global todas as pessoas encontradas no sistema. Cada pessoa é delimitada no algoritmo por um retângulo, e o primeiro passo é identificar o centro dele (linha 15).

A comparação entre a pessoa identificada no quadro atual com as pessoas do vetor global é realizada através da distância euclidiana (linha 19) e ocorre a fim de encontrar a pessoa mais próxima já identificada e armazenada no vetor global. Ao identificar a pessoa mais próxima, faz-se a verificação se a distância é menor ou maior que 100 (cem) (linha 20). O número 100 (cem) foi obtido empiricamente observando os quadros e a contagem. Caso a distância entre a pessoa identificada no quadro atual e a pessoa identificada no quadro anterior for maior que 100 (cem), significa que não existe uma pessoa tão próxima que possa se pressupor que é a mesma. Assim é definido que é uma nova pessoa identificada e dessa forma é realizado um

novo registro no vetor global de pessoas. Caso a distância for menor que 100 (cem), significa que existe uma pessoa tão próxima que pode ser a mesma pessoa, então pressupõe-se que é a mesma pessoa. Assim o sistema atualiza a posição da pessoa identificada no vetor global através da função (`updatePositions()`) (linha 30). Com a posição atualizada, realiza-se o processo de contagem através da função (`count()`) (linha 31) como exemplificado na Figura 4.7.

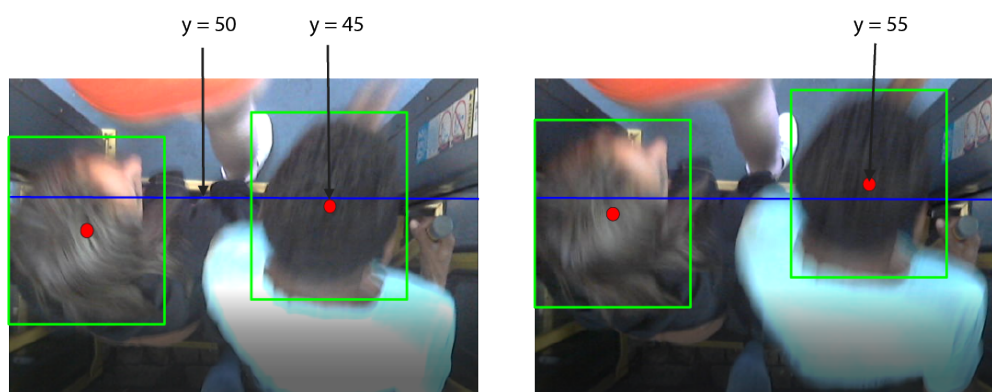


Figura 4.7 – Comparação de Contagem, Fonte: Próprio Autor.

Na comparação, o objetivo é identificar o embarque ou desembarque da pessoa. A função `count()` compara os pontos centrais de uma possível pessoa identificada através da função (`onOppositeSides()`) (linha 55). A contagem é realizada quando a pessoa cruza uma linha limítrofe definida (linha 6). Se no quadro anterior o ponto está abaixo da linha de limítrofe, e no quadro atual o ponto está na mesma linha ou acima é realizada a contagem de uma entrada. Se no quadro anterior o ponto está acima da linha limite, e no quadro atual está na mesma linha ou abaixo é realizada a contagem de uma saída.

4.3 ALGORITMO DE AGREGAÇÃO

Conforme mencionado na Seção 4.2, existe uma comunicação entre os módulos de Contagem acoplados nas portas do ônibus com o Módulo Agregador. No Algoritmo 4.2, apresentam-se as estratégias utilizadas pelo Módulo de Agregação para gerenciar as informações de contagem obtida dos módulos de contagem, e para enviar os dados de contagem para uma rede externa. Para um melhor entendimento o fluxo de funcionamento do algoritmo pode ser visto na Figura 4.8. O algoritmo de agregação é executado no módulo de agregação e opera na camada de Tratamento. O módulo de agregação é o hardware com poder de processamento e comunicação, assim ainda é responsável por enviar os dados agregados para uma rede externa ao ônibus. Esse processo de transmissão de dados é operado na camada de interface de comunicação.

Algorithm 4.2: Algoritmo de Agregação

```

1  begin
2    Módulo de Agregação recebe os dados dos módulos de Contagem
3    listaDeParadas = [];
4    listaDeMensagens = [];
5    Function agregacao(stringContagem)
6      cabecalho = stringContagem.substring(0, stringContagem.lastIndexOf("*"));
7      mensagem = stringContagem.substring(stringContagem.lastIndexOf("*"),
        stringContagem.length)
8      idParada = mensagem.substring(0, 3);
9      idPorta = mensagem.substring(3, 3);
10     numeroEntradas = mensagem.substring(6, 3);
11     numeroSaidas = mensagem.substring(9, 3);
12     idMensagem = cabecalho.substring(0, cabecalho.indexOf("*"));
13     paradaExiste = false;
14     paradaDuplicada = false;
15     Percorre por todas paradas armazenadas anteriormente
16     for cada  $u \in \text{listaDeParadas}$  do
17       if  $\text{idParada} == u.\text{idParada}$  then
18         paradaExiste = true;
19         for each  $m \in \text{listaDeMensagens}$  do
20           if  $\text{idMensagem} == u.\text{idMensagem}$  then
21             | paradaDuplicada = true;
22           end
23         end
24         if  $\text{paradaDuplicada} == \text{false}$  then
25           |  $u.\text{numeroEntradas} += \text{numeroEntradas}$ ;
26           |  $u.\text{numeroSaidas} += \text{numeroSaidas}$ ;
27           |  $\text{totalNoMomento} += \text{numeroEntradas} - \text{numeroSaidas}$ ;
28         end
29       end
30     end
31     if  $\text{paradaExiste} == \text{false}$  then
32       |  $\text{listaDeParadas.add}(\text{idParada}, \text{numeroEntradas}, \text{numeroSaidas})$ ;
33     end
34      $\text{listaDeMensagens.add}(\text{idMensagem}, \text{idParada}, \text{numeroEntradas},$ 
35        $\text{numeroSaidas})$ ;
36      $\text{enviaConfirmacao}(\text{"ACK\_Contagem"} + \text{cabecalho})$ ;
37   end
38   Function loop()
39      $\text{stringContagem} = \text{contabilizacao}()$ ;
40     if  $(\text{latitude} > n \text{ and } \text{latitude} < o) \text{ and } (\text{longitude} > p \text{ and } \text{longitude} < q)$  then
41       |  $\text{enviaContagemFinal}(\text{listaDeMensagens})$ ;
42     end
43     else
44       |  $\text{agregacao}(\text{stringContagem})$ ;
45     end
46   end

```

O módulo de Agregação mantém uma lista das paradas realizadas pelo ônibus durante o percurso. Quando o ônibus inicia a rota essa lista está vazia, sendo incrementada a cada parada. A informação é estruturada em uma *string* conforme apresentado na Figura 4.9.

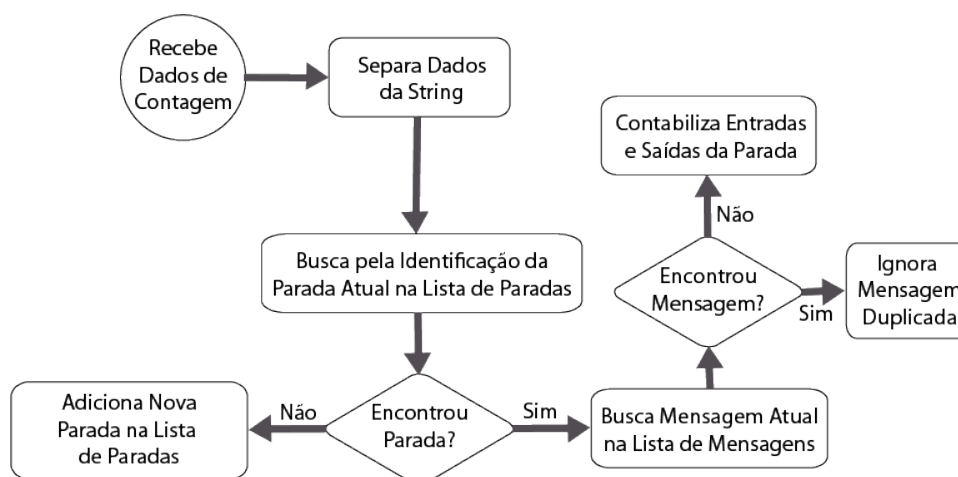


Figura 4.8 – Algoritmo de Agregação de Contagem das Portas, Fonte: Próprio Autor.

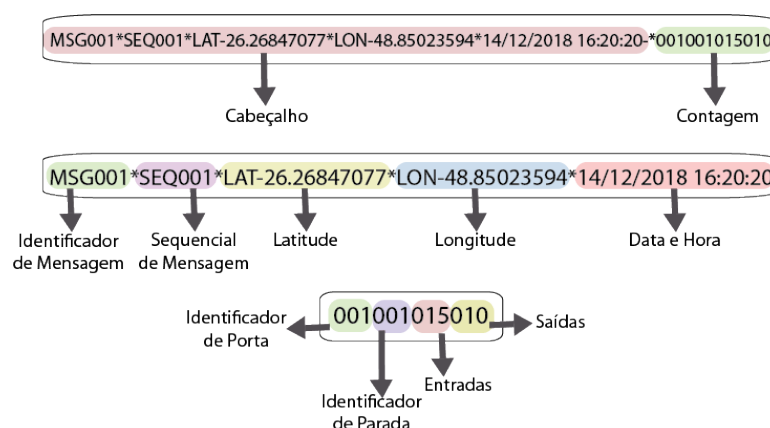


Figura 4.9 – String Algoritmo de Agregação de Contagem das Portas, Fonte: Próprio Autor.

Na Figura 4.9 estão representados 9 (nove) dados, divididos em um cabeçalho e o corpo da mensagem. O cabeçalho é composto por: Identificador da Mensagem (MSG001), Número Sequencial da Mensagem (SEQ001), Latitude (LAT-26.26847077), Longitude (LON-48.85023594) e Data com Hora da Parada (14/12/2018 16:20:20). O corpo da mensagem é composto por 4 tipos de dados: Identificação da Porta (001), Identificação da Parada (001), Número de Entradas (015), Número de Saídas (010). No conteúdo do pacote, definiu-se que a cada 3 (três) dígitos, tem-se uma variável.

O módulo de Contagem da porta, após realizar a contagem das pessoas, envia para o módulo de Agregação uma *string* com os dados da contagem de cada parada. Os dados recebidos na *string* "stringContagem"(linha 38) são separados entre cabeçalho e mensagem (linhas 6 e 7). Isso é necessário para que os dados de entrada e saída das diversas portas sejam somados, para obter o número total de passageiros que entraram ou saíram do ônibus em cada parada, conforme detalhado na Figura 4.10.

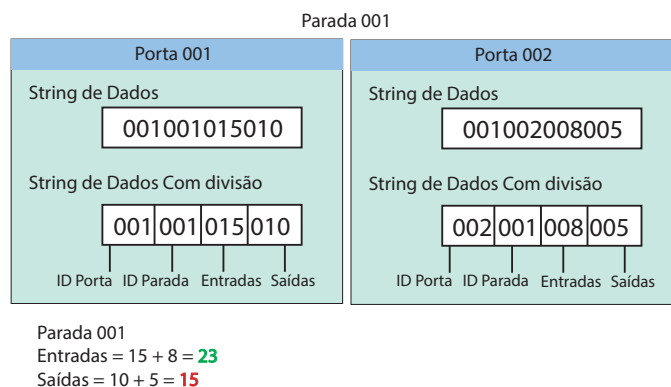


Figura 4.10 – Agregação da String de Contagem das Portas, Fonte: Próprio Autor.

Utiliza-se uma variável de controle, "paradaExiste"(linha 13), para identificar se a parada recebida já existe ou não. Para fins de exemplificação do uso da variável de controle, suponha o seguinte cenário: Um ônibus efetua uma parada e cada porta realiza a contagem individualmente gerando a *String* com os 4 (quatro) dados acima mencionados (dados da parada). Cada porta envia seus dados para o Módulo Agregador. Com isso, o registro da parada na lista de paradas é atualizado. Contudo, quando uma *string* com a informação de uma porta é recebida pelo agregador, o mesmo verifica se essa parada já foi registrada por outra porta. Caso sim, somente os números de entrada e saída são agregados, caso contrário, deve ser registrada a nova parada na lista.

Justifica-se emprego do mecanismo de controle do identificador de parada pelo fato que, as portas podem fechar em tempos diferentes ou a rede pode entregar os dados ao módulo Agregador em tempos não concomitantes. Caso nenhum identificador de parada seja encontrado na lista (linha 31), o identificador recebido é adicionado à lista (linha 32). O Agregador envia uma mensagem de confirmação para o módulo de Contagem (linha 35), avisando que os dados foram recebidos.

Quando o módulo de Contagem conclui o processamento e envia os dados para o Agregador, ele fica aguardando a confirmação do recebimento (ACK Contagem). Isso acontece para garantir que os dados foram recebidos e registrados pelo agregador. Nesse momento, o microcontrolador onde está o módulo de contagem pode eliminar os vídeos. Caso essa confirmação não retorne dentro de um período de tempo, o sistema assume que houve perda na comunicação e os dados são reenviados. O período de tempo é ajustável de acordo com o administrador da operação.

Pode ocorrer que a mensagem tenha chegado até o agregador, porém no momento do retorno de confirmação, a conexão tenha se perdido e o Ack não alcance ao módulo de Contagem. Para evitar que ocorra duplicação da contagem, todas as mensagens que chegam ao agregador são guardadas em uma lista (linha 34) com os

dados de parada e um identificador de mensagem. Antes de realizar a agregação é realizada uma busca na lista de mensagens (linhas 19 até 23) e comparado o identificador da mensagem atual com cada identificador da lista para garantir que aquela mensagem ainda não tenha sido armazenada. Caso a mensagem seja encontrada na lista (linha 20), um verificador de parada duplicada é atribuído com o valor *true* (linha 21). Dessa forma, se a parada não for duplicada (linha 24), a agregação ocorre normalmente. Caso contrário o sistema prossegue sem agregar e reenvia a mensagem de confirmação (linha 35).

Ao término da rota, por medida de segurança, para garantir a confiabilidade da contagem e a integridade da transmissão do módulo Agregador, os dados armazenados sobre a contagem no módulo Agregador são enviados para o servidor externo no formato de uma única mensagem, contendo todas as mensagens armazenadas no decorrer da rota (linha 40). No servidor, pode-se realizar uma conferência com as informações de contagem recebidas durante o percurso do ônibus a fim de encontrar possíveis divergências, por exemplo, da perda de alguma mensagem.

4.4 CONCLUSÕES DO CAPÍTULO

Neste capítulo foi apresentada uma abordagem para contagem de passageiros em transporte público urbano, especificamente, ônibus. A solução foi modelada na forma de uma arquitetura para coletar, tratar e transmitir os dados capturados através de câmeras instaladas no ônibus. A arquitetura é dividida em lógica e de implementação. A arquitetura lógica é composta pelas camadas de Percepção, Tratamento, Interface de Comunicação e Dados e Aplicação. A arquitetura de implementação é composta pelos módulos de Acionamento e Detecção, Contagem, Agregação e Aplicação.

Um algoritmo com base em Visão Computacional foi desenvolvido para atuar em microcontroladores embarcados no ônibus. Através da camada de Percepção, o sistema coleta as informações dos passageiros, e em seguida, realiza a contagem através da camada de Tratamento, onde opera o algoritmo de contagem baseado em Visão Computacional. Por fim, transmite os dados gerados para um Módulo Agregador, afim de unificar a contagem de todas as portas.

O Módulo Agregador, ao receber as informações do Módulo de Contagem, identifica outras contagens na mesma parada e soma as entradas e saídas através de um algoritmo de agregação desenvolvido. A transmissão das informações do Módulo de Contagem para o Módulo Agregador são realizadas via protocolo CoAP sob uma rede IEEE 802.11 b/g/n. O algoritmo de agregação desenvolvido, ao final do processo de unificação dos dados, realiza a transmissão dos dados agregados para um servidor externo via protocolo MQTT sob uma rede GPRS. A agregação dos dados realizadas

pelo módulo agregador opera na camada de Tratamento, enquanto a transmissão dos dados ocorre na camada de Interface de Comunicação.

Visando assegurar que a empresa receba as informações corretas sobre a contagem, definiu-se que ao final do percurso do ônibus, o Módulo Agregador envia para o servidor externo, uma mensagem contendo o total de embarques e desembarques em cada parada. O Módulo Agregador conhece o final da rota, por meio de coordenadas geográficas obtidas de um dispositivo do tipo GPS. Por fim, o Módulo Agregador deve receber uma mensagem de confirmação de recebimento do servidor externo. Esta última confirmação é garantida pelo uso do protocolo MQTT de QoS nível 1 implementado no Agregador.

5 EXPERIMENTOS E RESULTADOS

Nesta seção descreve-se os experimentos e os resultados obtidos com a solução proposta no Capítulo 4. Neste capítulo apresenta-se: (i) descrição dos cenários de testes; (ii) resultados com o algoritmo de contagem de passageiros e com o algoritmo de agregação e; (iii) análise dos resultados decorrentes dos experimentos.

5.1 AMBIENTE DE TESTES

No cenário alvo deste trabalho (descrito na Seção 1.1), um dos grandes desafios é a dificuldade para realizar a contagem de pessoas em face do fluxo significativo de passageiros que podem entrar ou sair do ônibus ao mesmo tempo. Neste contexto, identificou-se a Visão Computacional como uma ferramenta adequada para a detecção e contagem de passageiros. Portanto, câmeras foram instaladas no ônibus para a captura de imagens do fluxo de entrada e saída de passageiros pelas portas. A especificação dos equipamentos utilizados na implementação consta no Anexo A. O desempenho do algoritmo para a detecção e contagem de passageiros pode sofrer variações em função das características do cenário alvo. Alguns fatores podem contribuir para essas variações como, o elevado número de pessoas que podem entrar ou sair do ônibus ao mesmo tempo, variações na iluminação do ônibus decorrentes da abertura e fechamento das portas, condições climáticas ou hora do dia, desconexões do sistema de comunicação causados por ruídos do motor do ônibus, interferências de outras redes e oclusão de pessoas. Considerando-se os fatores supracitados, que influenciam no desempenho da solução proposta, identificou-se as métricas descritas a seguir como as que melhor se adequam à avaliação de desempenho desta solução:

- Percentual de Acerto: número real de pessoas que entraram/saíram do ônibus, comparado com a contagem efetuada pelo sistema.
- Falsos Positivos: pessoas contabilizadas pelo sistema incorretamente, ou seja, o sistema contabilizou uma pessoa que não existe.
- Tempo de Processamento: tempo de processamento do algoritmo para contabilizar cada vídeo no microcontrolador.
- Sobrecarga: número de pacotes de contagem enviados pelo sistema/número total de mensagens geradas na rede.
- Perda de Pacotes: número de pacotes enviados pela origem (microcontrolador no ônibus)/ número de pacotes recebidos pelo destino (servidor).

Os cenários utilizados nos experimentos, os resultados obtidos e a análise desses resultados serão descritos na seção seguinte.

5.2 EXPERIMENTOS E ANÁLISE DE RESULTADOS

Nesta seção, descrevem-se os cenários modelados para avaliação do algoritmo de contagem de passageiros e os resultados obtidos. Nos testes, avaliou-se a eficácia da contagem com a arquitetura proposta (Figura 4.1) no Capítulo 4 .

5.2.1 Análise do Algoritmo de Contagem de Passageiros

Os testes apresentados a seguir foram realizados nos ônibus da Empresa Transtusa na cidade de Joinville-SC. Nos experimentos, avaliou-se o potencial do algoritmo de contagem de pessoas no cenário alvo com base nos classificadores de Haar (VIOLA; JONES, 2001) e HOG (DALAL; TRIGGS, 2005).

5.2.1.1 Captura de Vídeos

Para a gravação dos vídeos, utilizou-se uma câmera do tipo Webcam com capacidade de captura de imagens HD (Alta Definição - 1280 x 720 pixels). Os vídeos foram recebidos e processados por um microcontrolador do tipo Raspberry Pi. As imagens foram gravadas no ônibus de número 1118 disponibilizado pela empresa. Este ônibus, diariamente percorre a rota de número 1360 entre os terminais Norte e a região de Pirabeiraba na cidade de Joinville-SC. Realizou-se testes com a câmera posicionada de duas formas: (i) no centro do teto do ônibus (entre uma lateral e outra do ônibus como pode ser visto na Figura 5.1), no ângulo de 45° em relação ao teto e voltada para a porta e; (ii) na posição zenital acima da porta com a câmera voltada para o chão como apresentado na Figura 5.2.

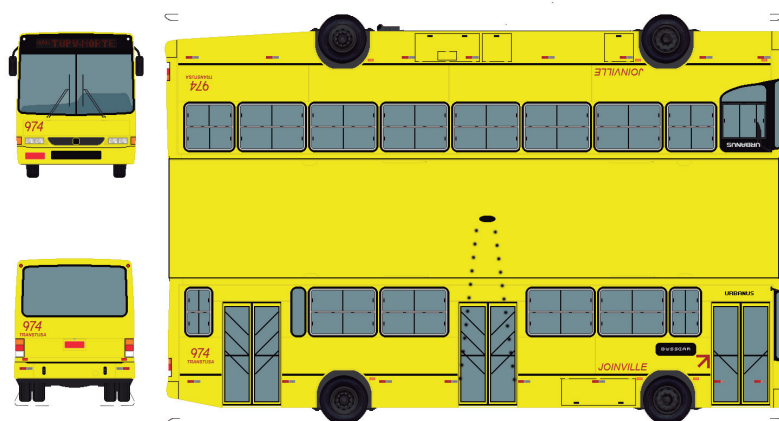


Figura 5.1 – Câmera na Posição com Ângulo de 45° Dentro do Ônibus, Fonte: Próprio Autor

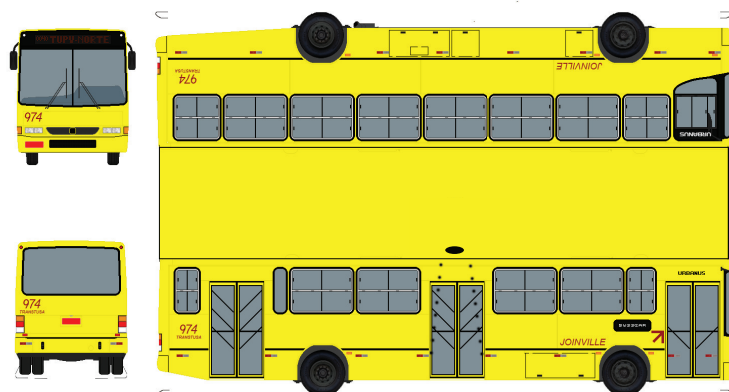


Figura 5.2 – Câmera na Posição Zenital Dentro do Ônibus, Fonte: Próprio Autor

Para a realização dos experimentos foram gravados vídeos em condições variadas de clima e horário. Especificamente, foram gravados 5 (cinco) vídeos na posição zenital e 2 (dois) com a câmera em um ângulo de 45° . Os vídeos apresentam as entradas e as saídas de passageiros. Ressalta-se que as rotas são as mesmas, a diferença está no período do dia. Os vídeos foram executados com uma taxa de 30 quadros por segundo, pois é a qualidade máxima que a câmera utilizada permite. Após a gravação dos vídeos, as imagens relacionadas aos embarques e aos desembarques foram separadas dos vídeos originais. Cada vídeo gerou um arquivo com 35 MB em média, totalizando 490 MB. Cada vídeo possui em média 52 segundos, variando de 28 segundos a 90 segundos, totalizando 12 minutos e 11 segundos de gravação. Na Tabela 5.1 apresenta-se uma descrição detalhada dos vídeos coletados e já separados dos vídeos originais. Os vídeos foram numerados de 1 (um) a 14 (quatorze) e classificados em relação a posição da câmera, ao período do dia em que ocorreu a captura e ao sentido do passageiro (embarque ou desembarque).

Tabela 5.1 – Classificação dos Vídeos Coletados

Vídeo	Pessoas	Diurno	Noturno	Posição	Sentido
Vídeo 1	27	X		Zenital	Embarque
Vídeo 2	21	X		Zenital	Embarque
Vídeo 3	18	X		Zenital	Embarque
Vídeo 4	25		X	Zenital	Embarque
Vídeo 5	20		X	Zenital	Embarque
Vídeo 6	28	X		Zenital	Desembarque
Vídeo 7	25	X		Zenital	Desembarque
Vídeo 8	30	X		Zenital	Desembarque
Vídeo 9	27		X	Zenital	Desembarque
Vídeo 10	22		X	Zenital	Desembarque
Câmera a 45°					
Vídeo 11	22	X		45°	Embarque
Vídeo 12	18		X	45°	Embarque
Vídeo 13	21	X		45°	Desembarque
Vídeo 14	32		X	45°	Desembarque

5.2.2 Resultados com o Algoritmo de Contagem

Nos experimentos, avaliou-se o algoritmo de contagem com base em duas técnicas para detecção de pessoas: Haar Likes e HOG. Para a realização dos experimentos, utilizou-se 1400 imagens positivas e 1400 negativas. As imagens, tanto positivas quanto negativas, foram obtidas dos vídeos reais coletados nos ônibus em meio às rotas. As positivas, foram feitas através da identificação das pessoas nos quadros e feitos recortes das imagens reais. As imagens negativas foram retiradas dos mesmos quadros, porém com recortes de áreas que não fazem parte das pessoas, como corrimão, escada, chão, assim como partes que não podem ser contabilizadas, por exemplo, mochilas, mãos e chapéus. Para treinamento essas imagens foram rotuladas através da separação em pastas distintas.

Cada imagem positiva foi comparada com todas as negativas gerando um mapeamento de 1400 vetores que foram utilizados para treinamento. No treinamento, aplicou-se 5000 combinações entre imagens positivas e 2500 combinações entre imagens negativas. Cada processo de treinamento foi realizado em 10 estágios a fim de otimizar o resultado. A cada estágio, repete-se o processo de comparação. Foram realizados 10 estágios, devido a limitação tecnológica do hardware (computador) utilizado para o treinamento.

Na Tabela 5.2, apresentam-se os resultados obtidos na contagem em relação ao tempo de processamento no microcontrolador embarcado no ônibus. Na tabela constam, o número de pessoas que entraram ou saíram do ônibus (Coluna Pessoas), o tempo de processamento em segundos (Coluna Tempo) e o número de quadros de cada gravação (Coluna Quadros). A técnica de Haar Likes foi aplicada neste experimento.

Tabela 5.2 – Resultados da Contagem (com Haar)

Vídeo	Pessoas	Tempo	Quadros
Vídeo 1	27	99	949
Vídeo 2	21	43	444
Vídeo 3	18	37	387
Vídeo 4	25	85	867
Vídeo 5	20	42	431
Vídeo 6	28	61	634
Vídeo 7	25	52	524
Vídeo 8	30	81	843
Vídeo 9	27	70	732
Vídeo 10	22	38	412
Câmera a 45°			
Vídeo 11	22	33	326
Vídeo 12	18	150	847
Vídeo 13	21	29	284
Vídeo 14	32	160	1578

Os resultados do tempo de processamento apresentado na Tabela 5.2, com Haar, indicam que um processamento em tempo real pode ser realizado no ônibus. O tempo de processamento foi maior nos vídeos com câmera em posição no ângulo de 45°, no período noturno (vídeos 12 e 14). O tempo de processamento é maior, pois as imagens necessitam de uma compensação de iluminação por estarem voltadas para porta e consequentemente fora do ônibus.

Na Figura 5.3, apresenta-se o percentual de acertos referentes ao embarque de passageiros com a câmera na posição zenital, e utilizou-se Haar com o algoritmo de contagem (descrito na Seção 4.2 do Capítulo 4).

No embarque diurno (Vídeos 1, 2 e 3), o percentual de acertos foi de 85,18% com o vídeo 1 e sem falsos positivos, enquanto no vídeo 2, o percentual de acertos foi de 85%, com 4% de falsos positivos. O vídeo 3 apresentou 83% de acertos, com 5% de falsos positivos.

Nos vídeos referentes ao embarque noturno (4 e 5), o percentual de acertos ficou em 84% com o vídeo 4, e 85% com o vídeo 5, ambos sem falsos positivos. Considera-se que percentual de acertos, em média 85%, resultante do emprego do algoritmo de Haar, é justificado pela complexidade para a obtenção de imagens de qualidade, dado as características do ambiente. Alguns vídeos apresentaram um número elevado de falsos positivos, por exemplo, 5% com o vídeo 3. Este número elevado de falsos positivos, decorre da dificuldade de identificação das pessoas na imagem por estarem muito próximas à câmera.

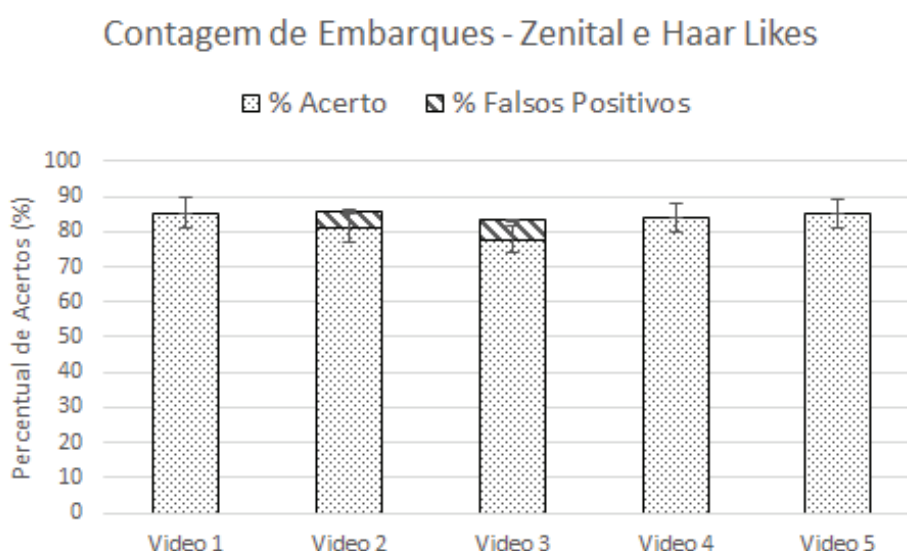


Figura 5.3 – Contagem no Embarque com Câmera em Posição Zenital e Haar

Na Figura 5.4, apresenta-se os resultados obtidos com o algoritmo de contagem referentes ao desembarque de passageiros com a câmera posicionada em modo

zenital. No desembarque diurno (Vídeos 6, 7 e 8), o percentual obtido foi 85,71% sem falsos positivos para o vídeo 6. Com o vídeo 8 o algoritmo de contagem alcançou um percentual de acertos de 86%, porém 3,33% é de falsos positivos, que diminui o percentual real para 83,33%. O número alto de falsos positivos encontrados nos vídeos 7 e 8, ocorrem nas situações em que a pessoa está portando uma mochila e entra muito rápido no ônibus. Nessas situações, a pessoa possui dois blocos de cores diferentes, cabeça e roupa ou então cabeça e mochila. Dessa forma o sistema contabilizou a pessoa duas vezes.

No desembarque noturno (Vídeos 9 e 10), o percentual de acertos foi de 85,81% para o vídeo 9, com nenhum falso positivo e 86,3% de acertos no vídeo 10, com 4,5% de falsos positivos. Constatou-se, que o número elevado de falsos positivos, por exemplo, 4,5% no processamento do vídeo 10, é decorrente da dificuldade de identificação das pessoas na imagem por estarem próximas à câmera. Neste caso, o algoritmo de contagem não conseguiu diferenciar adequadamente pessoas de objetos.

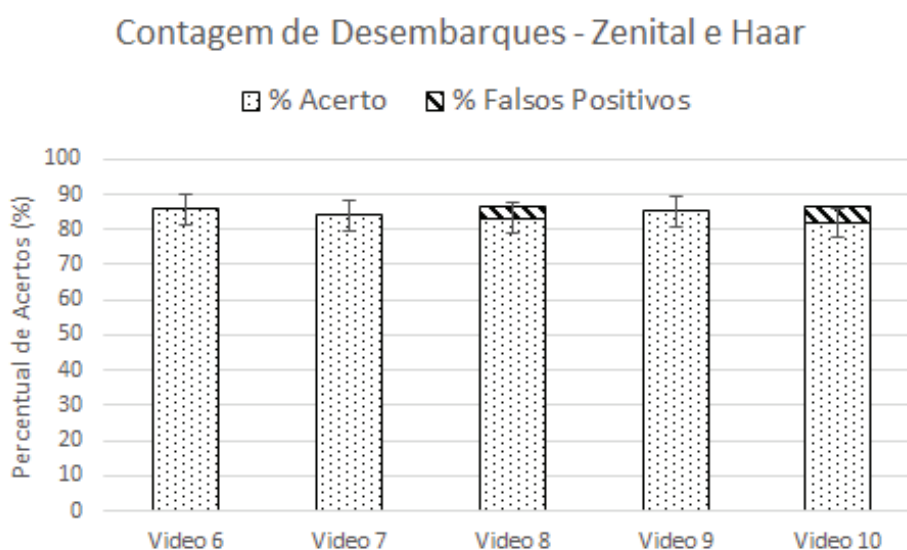


Figura 5.4 – Contagem no Desembarque com Câmera em Posição Zenital e Haar

O percentual de acertos obtidos nos experimentos com câmera em posição zenital ficaram abaixo de 90%. Este resultado, justifica-se pela pouca distância entre os passageiros (entrada e saída) e a câmera. Quanto mais próximo da câmera está a pessoa, maior é o espaço ocupado por ela na imagem, resultando na dificuldade de assimilar formas. Na literatura, conforme apresentado por (LI et al., 2016) e (NALEPA; SZYMANEK; KAWULOK, 2015) (ver Capítulo 3), sugere-se utilizar câmeras bifocais com sensoreamento de profundidade para melhorar a detecção em ambientes com proximidade menor entre pessoas e câmera. Contudo, o uso de câmeras bifocais exige

maior poder de processamento para a execução das imagens, não sendo adequada para ser utilizada em hardware embarcado de baixa capacidade.

Para fins de comparação com os resultados apresentados com a câmera na posição zenital, mudou-se a câmera para a posição a 45° graus no ônibus (ver Figura 5.1). Nos experimentos apresentados a seguir, a técnica de Haar foi utilizada em conjunto com o algoritmo de contagem.

Na Figura 5.5, demonstra-se o percentual de acertos referente ao embarque de passageiros. No embarque diurno (Vídeo 11), o percentual de acertos foi de 50%, e 5% de falsos positivos. Na análise individual deste vídeo, verificou-se um alto índice de oclusão na imagem que não permitiu que o sistema identificasse algumas pessoas. Como a câmera está a 45°, obtém-se uma visão frontal e mais distante da porta nas imagens capturadas. Algumas pessoas passaram pela porta sem cruzar a linha limítrofe. A função dessa linha é a contagem, ou seja, ao cruzar por ela o sistema entende que uma pessoa entrou ou saiu no ônibus.

No embarque noturno (Vídeo 12), o percentual de acertos foi 55%, porém com mais falsos positivos, 11%. Na análise do vídeo 12, verificou-se a presença de sombras nas imagens. Neste caso, o sistema contabilizou erroneamente algumas sombras.

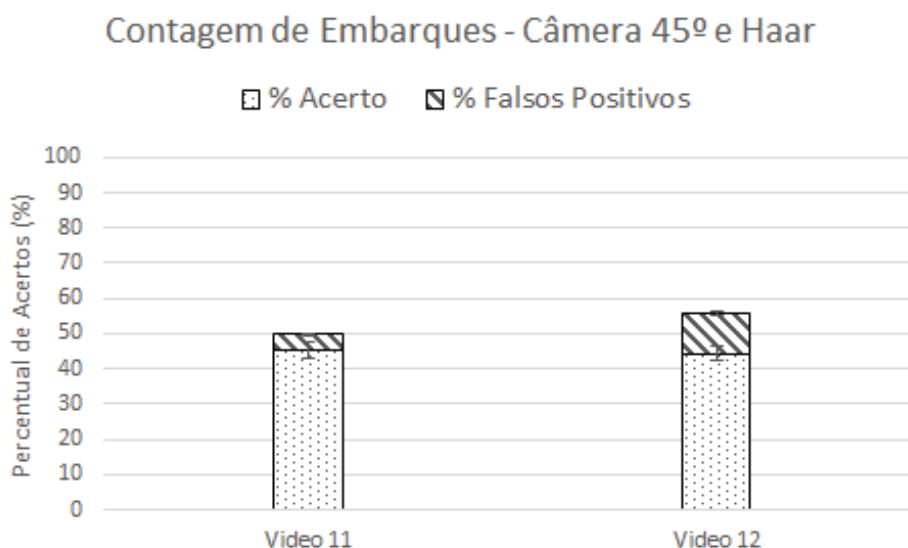


Figura 5.5 – Contagem no Embarque com Câmera em Posição 45° e Haar

Na Figura 5.6, apresenta-se o percentual de acertos referente ao desembarque de passageiros com a câmera a 45°. Neste cenário, tem-se os seguintes resultados: desembarque diurno (Vídeo 13), percentual de acertos de 47%, sem falsos positivos, e desembarque noturno (Vídeo 14), percentual de 46%, com 6% de falsos positivos.

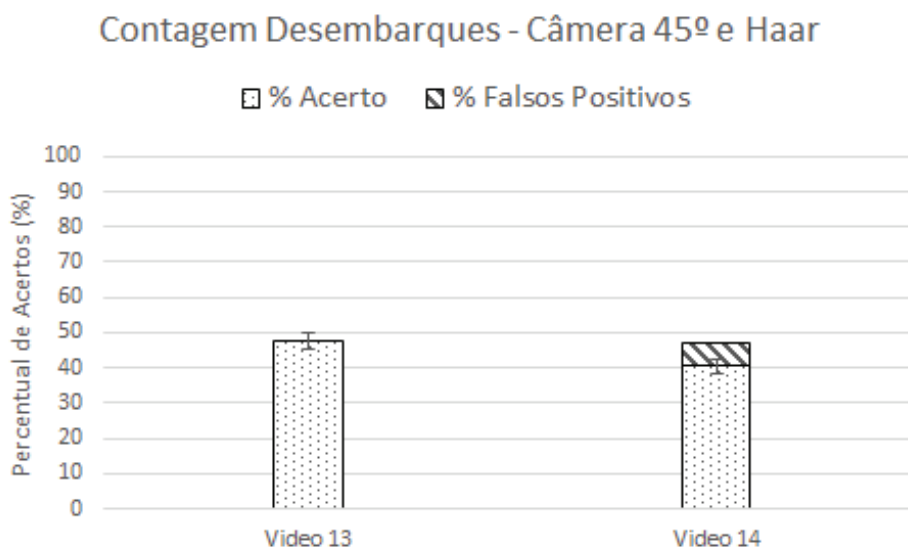


Figura 5.6 – Contagem no Desembarque com Câmera em Posição 45° e Haar

Com base na análise dos resultados com a câmera em posição a 45°, conclui-se que a posição da câmera favoreceu a presença de oclusões das pessoas, pelo fato da imagem ser coletada de frente e assim uma pessoa sobrepõe a quem vem atrás. Além disso, constatou-se que algumas pessoas saem da imagem pela lateral, sem cruzar pela linha limítrofe de contagem, pois a câmera abrange uma área além das escadas. Outra situação é, quando uma pessoa para no corredor se estiver no alcance da câmera, essa irá cobrir a cobertura da câmera.

Nos experimentos apresentados anteriormente, a técnica de Haar foi aplicada junto ao algoritmo de contagem para identificar as pessoas que passam pela porta. Visando realizar uma comparação com os resultados obtidos com a técnica Haar Likes, foram realizados experimentos com o mesmo ambiente de testes, porém escolheu-se a técnica de HOG. HOG foi a técnica escolhida por possuir uma ampla utilização na literatura para a identificação de pessoas.

5.2.2.1 Resultados com o Algoritmo de Contagem: HOG

Com a técnica de HOG, utilizou-se o treinamento padrão do OpenCV com um classificador SVM. Na Tabela 5.3, descrevem-se os resultados do tempo de processamento no microcontrolador com a contagem de passageiros.

Os resultados mostram que o tempo de processamento com a técnica de HOG no microcontrolador embarcado, foi maior quando comparado ao tempo obtido com a técnica de Haar (ver Tabela 5.2). HOG realiza um cálculo pixel a pixel para encontrar o histograma da imagem e detectar as bordas das cabeças, enquanto Haar otimiza essa busca através do cálculo da integral da imagem, resultando no aumento do tempo de

processamento.

Tabela 5.3 – Resultados da Contagem (com HOG)

Vídeo	Pessoas	Tempo	Quadros
Vídeo 1	27	1780	949
Vídeo 2	21	737	444
Vídeo 3	18	703	387
Vídeo 4	25	1439	867
Vídeo 5	20	756	431
Vídeo 6	28	1048	634
Vídeo 7	25	884	524
Vídeo 8	30	1355	843
Vídeo 9	27	1196	732
Vídeo 10	22	722	412
Câmera a 45°			
Vídeo 11	22	594	326
Vídeo 12	18	2250	847
Vídeo 13	21	495	284
Vídeo 14	32	2560	1578

Na Figura 5.7, mostra-se o efeito da técnica de HOG na contagem do embarque com a câmera na posição zenital. No embarque diurno (Vídeos 1, 2 e 3), o percentual de acertos alcançado foi de 70% para o vídeo 1, 71% para o vídeo 2 e 72% para o vídeo 3. Os falsos positivos para os vídeos 1, 2 e 3 foram 3%, 9% e 11%, respectivamente. No embarque noturno (Vídeos 4 e 5), o percentual de acertos foi de 64%, sem falsos positivos para o vídeo 4 e 70% de acertos, com 5% de falsos positivos para o vídeo 5.

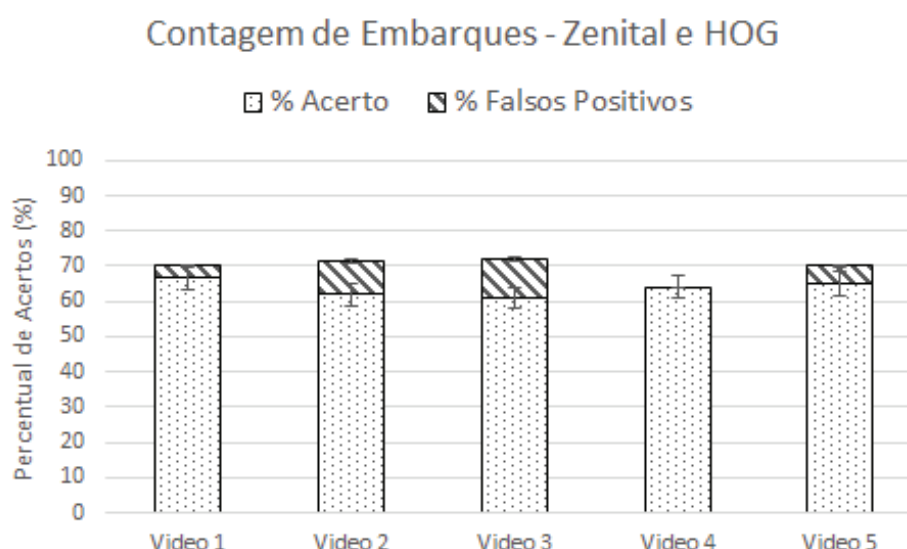


Figura 5.7 – Resultados da Contagem no Embarque com Câmera em Posição Zenital e HOG

O baixo percentual de acertos, utilizando-se a técnica de HOG junto ao algoritmo de contagem, deve-se ao emprego do classificador aplicado que utiliza um

treinamento padrão disponibilizado pelo OpenCV. Dessa forma, o treinamento não foi personalizado para o cenário alvo. Um treinamento personalizado não foi aplicado devido ao tempo de processamento requerido pela técnica de HOG ser muito elevado, tornando inviável a sua utilização em um sistema que exige um processamento embarcado e em tempo real.

Os desembarques (Figura 5.8) apresentaram percentuais de acertos igualmente baixos. O percentual de acertos com os vídeos diurnos (Vídeos 6, 7 e 8) ficou em torno de 70%. No desembarque noturno, também com a câmera na posição zenital (Vídeos 9 e 10) o percentual de acertos foi de aproximadamente 66%, considerando neste cálculo os falsos positivos.

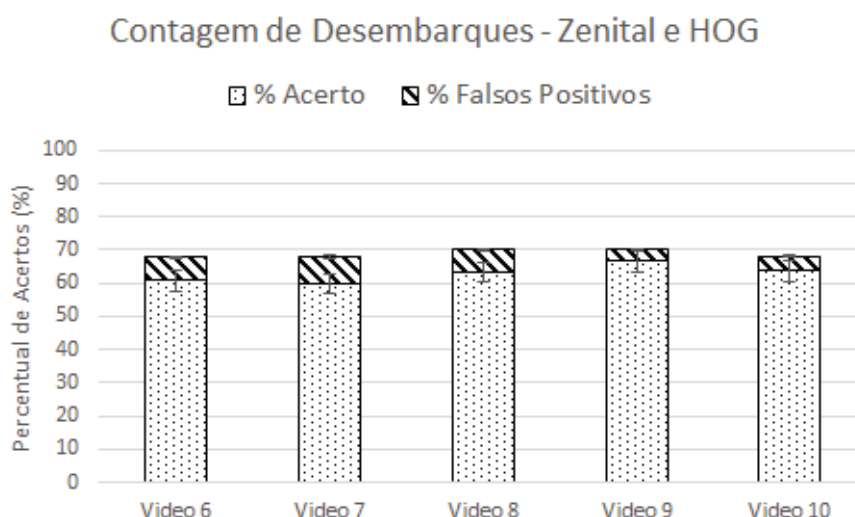


Figura 5.8 – Resultados HOG no Desembarque com Câmera em Posição Zenital

A Figura 5.9, ilustra os resultados da aplicação do algoritmo de contagem no embarque com a câmera na posição a 45°. A técnica de HOG não adicionou vantagem ao cenário avaliado, considerando-se a métrica percentual de acertos. Para o vídeo 11, o percentual de acertos foi de 45%, com 18% de falsos positivos e, para o vídeo 12, o percentual foi de 55%, com 22% de falsos positivos. Conclui-se com este resultado, que o posicionamento da câmera (posição na qual as pessoas são vistas em visão frontal), afetou o desempenho da técnica de HOG. Esta técnica apresenta maior dificuldade na identificação de pessoas, quando deve analisar maior variedade de formas e oclusões.

Tendo em vista que o percentual de acertos nos experimentos apresentados acima ficaram abaixo de 90%, investigou-se técnicas no contexto de *Deep Learning* utilizadas na literatura da área para o processamento de imagens. Escolheu-se a técnica *YOLOV3* porque vem sendo amplamente utilizada na detecção e classificação de objetos.

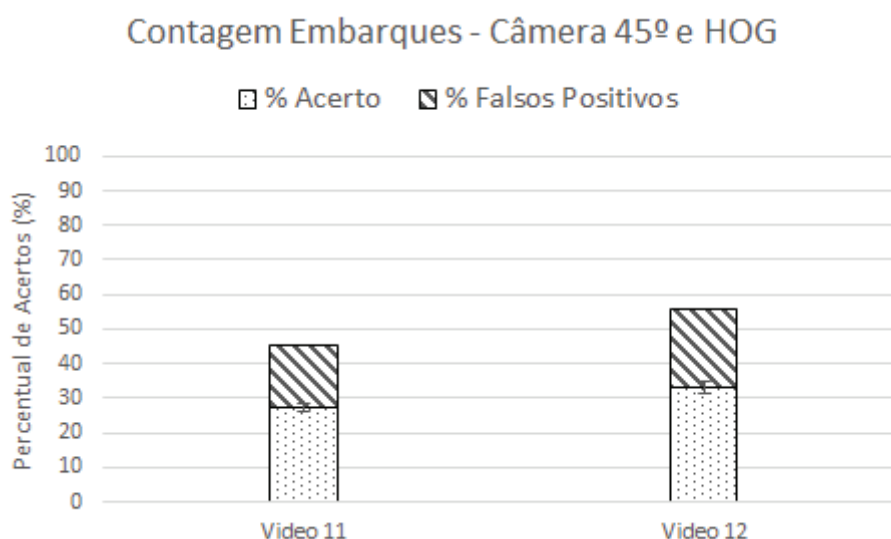


Figura 5.9 – Resultados da Contagem no Embarque com Câmera em Posição 45° e HOG

5.2.2.2 Resultados com o Algoritmo de Contagem: YOLOV3

Lin et al. (LIN et al., 2018), apresentaram um experimento comparando técnicas de detecção de objetos, e os resultados mostraram que YOLO obteve o melhor desempenho em menor tempo. Além disso, Redmon (REDMON; FARHADI, 2018) afirmam que YOLOV3 é um bom detector, rápido e com alta acurácia. Na Tabela 5.4, são apresentados os resultados do tempo de processamento utilizando o YOLOV3.

Tabela 5.4 – Resultados da Contagem (com YOLOV3)

Vídeo	Pessoas	Tempo	Quadros
Vídeo 1	27	2102	949
Vídeo 2	21	893	444
Vídeo 3	18	843	387
Vídeo 4	25	1715	867
Vídeo 5	20	864	431
Vídeo 6	28	1298	634
Vídeo 7	25	1012	524
Vídeo 8	30	1611	843
Vídeo 9	27	1479	732
Vídeo 10	22	835	412
Câmera a 45°			
Vídeo 11	22	762	326
Vídeo 12	18	1747	847
Vídeo 13	21	603	284
Vídeo 14	32	3074	1578

Os resultados apresentados na Tabela 5.4, mostram que o tempo de processamento do algoritmo de contagem com a técnica YOLOV3 foi maior no processamento dos vídeos, quando comparado com o algoritmo junto com as técnica de Haar e HOG. Isso acontece, pois a YOLOV3 utiliza 53 camadas convolucionais que exigem

um poder de processamento grande. Contudo, o percentual de acertos obtido foi maior conforme demonstrado nas Figuras 5.10 e 5.11.

Na Figura 5.10 é mostrado o efeito da YOLOV3 na contagem do embarque com a câmera na posição zenital. No embarque diurno (Vídeos 1, 2 e 3), o percentual de acertos foi de 92% (vídeo 1), sem falsos positivos, 90% (vídeo 2) e 5% de falsos positivos, e 88% (vídeo 3), também sem falsos positivos. No embarque noturno (Vídeos 4 e 5) o percentual de acertos foi de 88% para o vídeo 4, e 90% para o vídeo 5, sem falsos positivos para ambos os vídeos. Na Figura 5.11, demonstra-se o percentual de

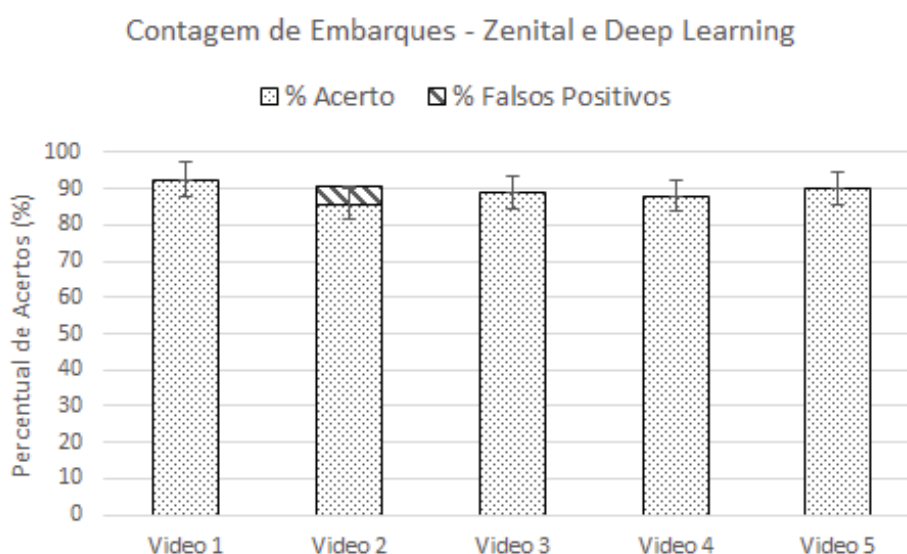


Figura 5.10 – Contagem no Embarque com Câmera em Posição Zenital e YOLOV3

acertos em relação ao desembarque capturados com a câmera em posição zenital. Por exemplo, nos vídeos diurnos (Vídeos 6, 7 e 8) o maior percentual de acerto obtido foi de 92% sem falsos positivos para o vídeo 1. No desembarque noturno (Vídeos 9 e 10) o percentual de acertos foi de 92% (vídeo 9) e 90% (vídeo 10), ambos sem falsos positivos.

Os resultados apresentados na Tabela 5.4, mostram que o tempo de processamento da contagem com a técnica YOLOV3 foi maior no processamento dos vídeos, quando comparado com o algoritmo junto com as técnica de Haar e HOG. Isso acontece, pois a YOLOV3 utiliza 53 camadas convolucionais que exigem um poder de processamento grande. Contudo, o percentual de acertos obtido foi maior conforme demonstrado nas Figuras 5.10 e 5.11.

5.2.3 Análise do Algoritmo de Agregação de Dados

Para avaliar o desempenho do algoritmo de agregação, foram realizados experimentos no quais os dados de contagem foram enviados entre os módulos Agregador

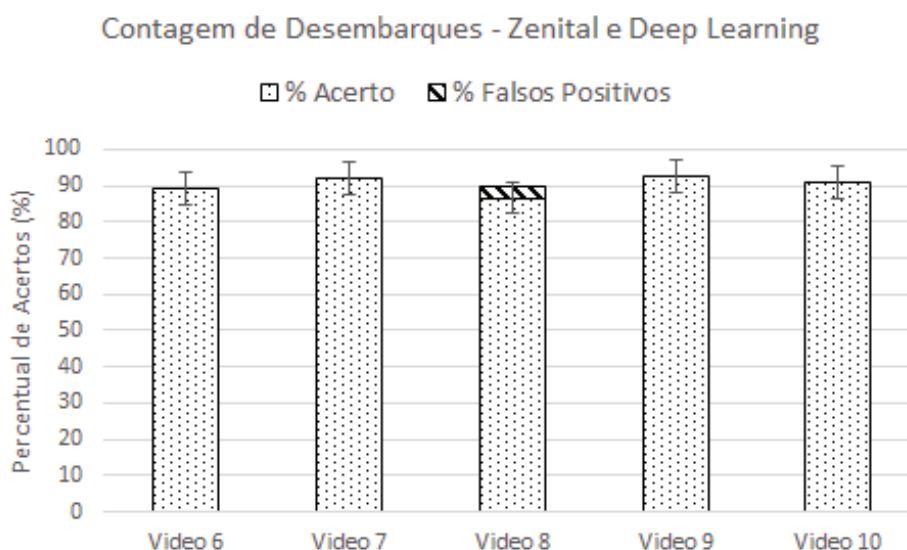


Figura 5.11 – Contagem no Desembarque com Câmera em Posição Zenital e YOLOV3

e de Contagem, e do módulo Agregador para um servidor externo. Utilizou-se um ônibus da empresa Transtusa como ambiente de testes.

O Módulo Agregador é composto por uma Placa NodeMCU ESP32 com uma placa AI Thinker que suporta a tecnologia GPRS e permite enviar dados por redes celulares. O Agregador foi implementado como *Access Point*, a fim de permitir que os Módulos de Contagem se conectem a ele.

No agregador foi implementado um *CoAP Server* para receber as mensagens referentes a contagem. O CoAP, conforme apresentado no Capítulo 2 é um protocolo de camada de aplicação desenvolvido no contexto da Internet das Coisas em substituição ao HTTP. No Módulo de Contagem foi implementado um *CoAP Client* para enviar as mensagens para o Agregador. Dessa forma ao terminar a contagem, o Módulo de Contagem envia os dados para o Módulo Agregador via protocolo CoAP. O CoAP foi escolhido como protocolo de aplicação, por rodar sob o protocolo de transporte UDP, tornando a comunicação mais leve e rápida.

O módulo Agregador necessita enviar os dados para uma rede externa. Neste sentido, foi implementado no agregador um Cliente *MQTT Publisher* de QoS nível 1, a fim de enviar as mensagens para um servidor externo. O MQTT com QoS de nível 1, garante que uma mensagem de reconhecimento seja enviada para garantir a confiabilidade das mensagens transmitidas pelo Agregador. O MQTT é um protocolo de camada de aplicação (ver Capítulo 2) e utiliza o protocolo de transporte TCP, fornecendo um mecanismo de confirmação do recebimento da mensagem pelo destino final.

Ao terminar uma contagem a cada fechamento de porta, o módulo de Contagem envia via protocolo CoAP os dados da contagem juntamente com as coordenadas obtidas via GPS para o Módulo Agregador. A comunicação entre os módulos Agregador e Contagem ocorre por meio de uma rede 802.11 b/g/n. O Módulo de Contagem, deve receber um pacote de confirmação (Ack) do Módulo Agregador, confirmando o recebimento dos dados. A mensagem do tipo Ack notifica o Módulo de Contagem que a contagem foi registrada pelo agregador, e os vídeos podem ser excluídos do microcontrolador Raspberry Pi. Contudo, destaca-se que outros microcontroladores podem ser utilizados como módulo de Contagem.

Com os dados de contagem das paradas processados e unificados, o Agregador pode enviá-los para um servidor externo. Nos experimentos, o envio para rede externa ao ônibus foi realizado através do protocolo MQTT sob uma rede celular do tipo GPRS. Para o recebimento das mensagens enviadas pelo Agregador para um servidor externo foi implementado um Broker CloudMQTT em um servidor na Internet. No módulo Agregador, que é composto por um NodeMCU ESP32, foi implementado um cliente *MQTT* para enviar as mensagens ao Broker. Os brokers do tipo MQTT estão bastante consolidados na Internet e oferecem maior facilidade de implementação aos usuários finais, quando comparado ao protocolo CoAP. Este fator motivou a escolha do MQTT como servidor externo.

A informação de contagem enviada para o agregador é textual e consiste em dados oriundos do processamento dos vídeos, realizado pelo equipamento Raspberry Pi. Cada linha de texto contém 82 bytes representando o identificador da mensagem, um número sequencial da mensagem, número de embarques e desembarques, identificador de porta, identificador da parada, posicionamento GPS, data e horário. Visando avaliar a escalabilidade da arquitetura de comunicação proposta, um número de pacotes (1, 5, 10 e 15) foram transmitidos entre os módulos de Contagem e Agregador, e entre o módulo Agregador e o servidor externo. Cada experimento foi executado 30 (trinta) vezes. As métricas avaliadas foram: perda de pacotes e atraso-fim-a-fim (envio dos dados de contagem).

Na Figura 5.12, apresenta-se os resultados do atraso-fim-a-fim entre o Módulo de Contagem e o Agregador. Durante os testes, uma instância do Wireshark (WIRESHARK.ORG, 2017) ficou em execução, capturando os pacotes que trafegaram na rede. Os pacotes capturados com o Wireshark foram analisados, visando calcular o atraso médio entre o Raspberry e o ESP32 na transmissão dos pacotes. A tecnologia de comunicação empregada nos experimentos foi o 802.11 b/g/n.

Como resultado, obteve-se um atraso médio (30 execuções) de 666ms para transmitir 1 (um) pacote, 674ms para transmitir 5 (cinco) pacotes, 682ms para transmitir 10 (dez) pacotes e 692ms para transmitir 15 (quinze) pacotes.

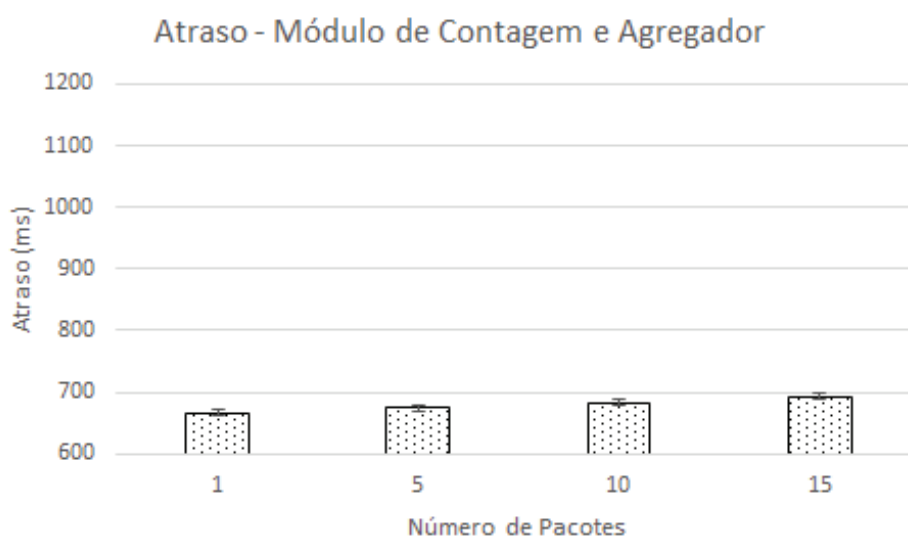


Figura 5.12 – Análise do Atraso Fim-a-Fim - Contagem e Agregador.

O atraso médio entre o Módulo de Contagem e o Agregador aumentou a medida que foram adicionados mais pacotes, pois o Agregador necessita enviar uma mensagem de confirmação (ACK) ao Módulo de Contagem, o que ocasiona uma fila de espera. Contudo, o atraso obtido é baixo, não interferindo no tratamento dos dados pelos módulos de Contagem e Agregador.

Na Figura 5.13, apresenta-se uma análise da perda de pacotes no mesmo ambiente de configuração apresentado na Figura 5.12. A tecnologia de comunicação empregada nos experimentos de comunicação entre os módulos dentro do ônibus foi o 802.11 b/g/n.

Na transmissão de 1 (um) e 5 (cinco) pacotes o percentual de perdas foi de 0,17% e o desvio padrão foi 0,07%. Na transmissão de 10 (dez) pacotes, o percentual de perdas foi de 0,17% com desvio padrão 0,08%. Com 15 pacotes o percentual de perdas foi de 0,18% com desvio padrão 0,08%. Através dos resultados obtidos, verificou-se que o mecanismo de comunicação proposto para atuar entre os módulos Contagem e Agregador, mostrou-se robusto em relação as perdas, mesmo no cenário com a transmissão de 15 (quinze) pacotes.

Na Figura 5.14, apresenta-se os resultados do atraso fim-a-fim entre o Módulo Agregador e um servidor externo. A tecnologia de comunicação empregada nos experimentos foi o GPRS.

Como resultado, obteve-se um atraso médio (30 execuções) de 1013ms para transmitir 1 (um) pacote, 1017ms para transmitir 5 (cinco) pacotes, 1019ms para transmitir 10 (dez) pacotes e 1022ms para transmitir 15 (quinze) pacotes.

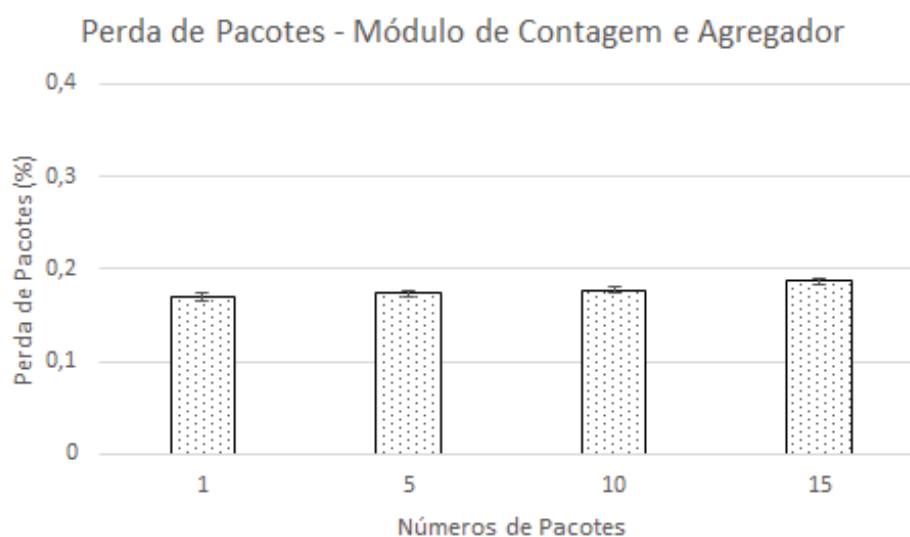


Figura 5.13 – Análise da Perda de Pacotes - Contagem e Agregador.

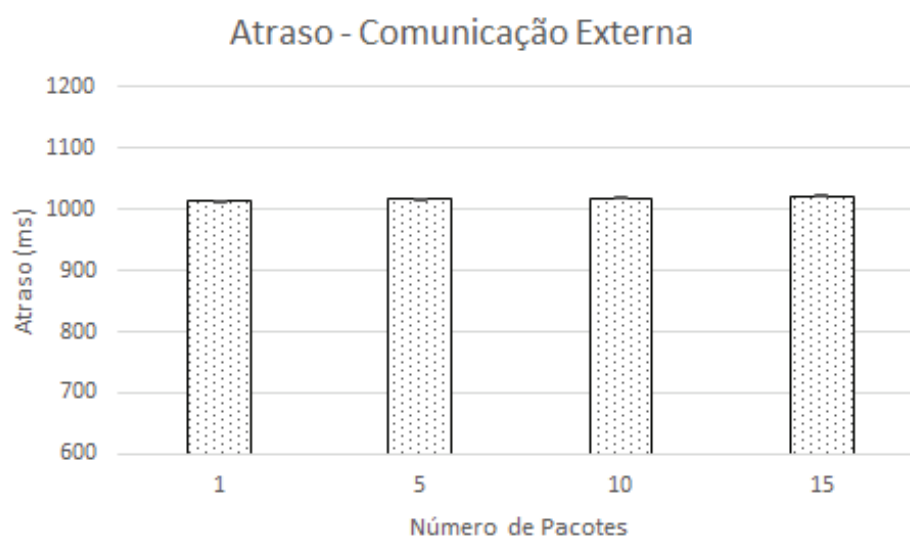


Figura 5.14 – Análise do Atraso Fim-a-Fim - Agregador e Servidor.

O atraso médio entre o Módulo Agregador e o servidor externo, foi maior quando comparado aos resultados do atraso apresentado na Figura 5.12. Este comportamento justifica-se, pela uso do protocolo de transporte TCP associado ao MQTT. Além disso, deve-se considerar o atraso do envio dos pacotes para o servidor externo via a rede GPRS. Outrossim, o atraso obtido é baixo, não interferindo no resultado final da comunicação.

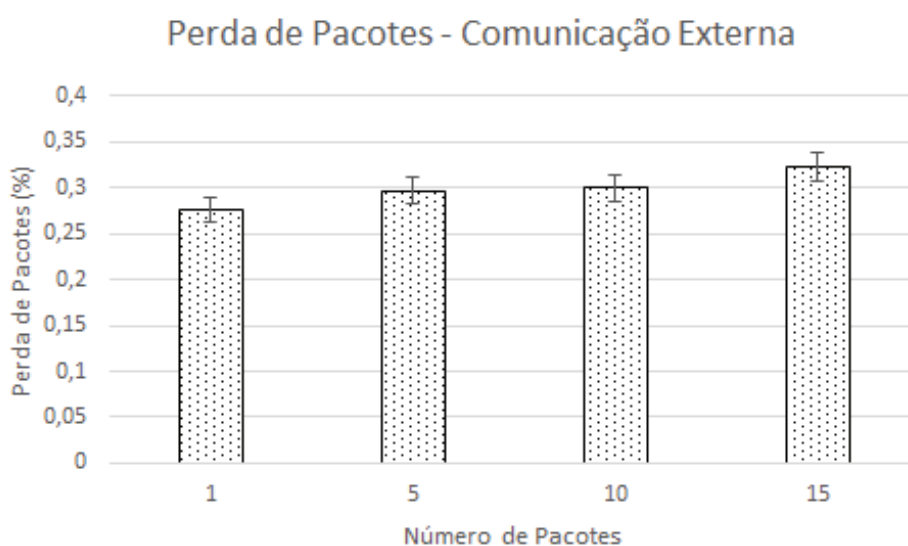


Figura 5.15 – Análise da Perda de Pacotes - Agregador e Servidor.

5.3 CONCLUSÕES DO CAPÍTULO

Neste capítulo foi apresentada a análise de desempenho da abordagem proposta para a contagem de passageiros em ônibus urbano coletivo. Foram realizados experimentos para avaliar o algoritmo para a contagem de passageiros e o algoritmo de agregação de dados.

Para avaliar o algoritmo de contagem no ônibus, mudou-se a posição da câmera e as técnicas de Visão Computacional. Adicionalmente, avaliou-se a técnica de *Deep Learning* YOLOV3 em relação a contagem de passageiros. Na avaliação da arquitetura para a comunicação entre microcontroladores e com a rede externa, variou-se o número de pacotes enviados na rede.

Nos experimentos, foram utilizadas 3 (três) técnicas diferentes: Haar, HOG e YOLOV3. Haar e HOG foram aplicadas no algoritmo de contagem de passageiros apresentado no Capítulo 4. Para aplicar a técnica YOLOV3, utilizou-se o algoritmo proposto por (REDMON et al., 2016). O percentual de acertos, sem falsos positivos apresentado pelo YOLOV3 foi o maior dentre as 3 técnicas. Porém, o tempo de processamento da contagem foi muito elevado, chegando a 35 minutos para processar um vídeo de 949 quadros (27 pessoas entrando no ônibus) e 30 FPS. HOG apresentou o pior desempenho, em torno de 70% de acertos, e tempo de processamento similar ao YOLOV3.

Conclui-se, que Haar apresentou o melhor resultado, pois alcançou um percentual de acertos de 85%, e um tempo médio de 412 segundos no processamento dos mesmos vídeos processados com YOLOV3 e HOG. O posicionamento da câmera

que apresentou melhor assertividade, foi a posição zenital. A câmera posicionada a 45%, gerou muita oclusão nas imagens. Destaca-se, que a contagem de passageiros no cenário alvo foi prejudicada devido ao fato das cabeças das pessoas passarem muito próximas às câmeras, ocupando um espaço grande na imagem, dificultando na identificação dos contornos.

Experimentos foram realizados para avaliar a arquitetura de comunicação implantada no ônibus. Nos experimentos, foram avaliados a perda de pacotes e o atraso fim-a-fim. Uma perda alta assim como um atraso alto podem ocasionar um congestionamento na fila de transmissão de dados e comprometer a confiabilidade da execução em tempo real do sistema. Os resultados mostraram que ao aumentar o número de pacotes, tanto a perda, quanto o atraso também aumentam. Apesar do aumento, a taxa de perda e o atraso não interferem negativamente na operacionalização do sistema. O comportamento foi similar nas transmissões entre o módulo de Comunicação e o Agregador e entre o Módulo Agregador e o servidor externo.

6 CONCLUSÕES FINAIS

Neste trabalho, apresentou-se uma abordagem de Visão Computacional com Internet das Coisas para disponibilizar informações que possam auxiliar no transporte público urbano, sobretudo informação na contagem dos passageiros.

No Capítulo 4 foi apresentado uma arquitetura para detecção e contagem de pessoas em ônibus. A abordagem apresenta uma estrutura para contagem, agregação e transmissão dos dados da contagem de passageiros. A contagem acontece utilizando técnicas de visão computacional. Após a contagem os dados de todas as portas são enviados para um outro microcontrolador e agregados, isso acontece a cada parada. Por fim, os dados agregados são enviados para um servidor externo ao ônibus. A comunicação acontece utilizando protocolos de internet das coisas.

Dentre as técnicas de Visão Computacional, escolheu-se a Características de Haar (CH) e o Histogramas de Gradientes Orientados (HOG). Para fins de comparação foi avaliada a técnica de *deep learning*, YOLOV3 na contagem de passageiros.

Nos experimentos apresentados no Capítulo 5, constatou-se que a técnica, Características de Haar (CH) apresentou o melhor resultado após a realização de treinamentos com um classificador Adaboost. Os resultados chegaram a 86%, enquanto, HOG alcançou 70%. Além disso, HOG apresentou problemas quanto ao longo tempo de processamento nos microcontroladores embarcados no ônibus. Com YOLOV3 os resultados da contagem chegaram a 92%, porém seu tempo de processamento foi muito superior ao Haar. Para utilização de YOLOV3 de forma precisa e com tempo de processamento baixo, é necessário uma placa de processamento auxiliar que comporte a execução de várias camadas convolucionais.

A arquitetura se apresentou como a maior contribuição deste trabalho, contudo, o processo de contagem, mais especificamente o algoritmo de contagem, necessita de uma investigação e testes mais apurados. Quando comparada aos trabalhos encontrados na literatura, nenhum apresentou uma arquitetura similar ao proposto no escopo deste trabalho. (BOREIKO; TESLYUK, 2016) apresentou uma estrutura, porém não leva em consideração o processamento embarcado, os vídeos são armazenados e enviados através de um modem 3G para a rede externa. Dessa forma, têm-se um custo elevado de transmissão pelo volume de dados.

Observou-se que o posicionamento da câmera, tanto a 45° em relação ao teto e voltada para a porta quanto em posição zenital, capta imagens muito próximas a cabeça das pessoas, resultando em oclusão e dificuldade na detecção das bordas dos corpos e cabeças das pessoas. Essa proximidade difere dos trabalhos encontrados na

literatura, em que a câmera é alocada mais distante ou possui sensor de profundidade. A proximidade das pessoas com a câmera faz com que a pessoa ocupe um espaço grande na imagem.

Com o objetivo de prover comunicação interna e externa no ônibus, foram implementados dois módulos, Agregador e Comunicação. A transmissão interna ocorre via rede 802.11 b/g/n e os dados são enviados através do protocolo CoAP.

Após a agregação dos dados das portas, os dados são transmitidos para um servidor na rede externa por meio do protocolo MQTT. Neste caso, a tecnologia de comunicação escolhida para a transmissão foi o GPRS por alcançar longas distâncias. O sistema identifica o término de uma rota, através do posicionamento GPS implementado em cada módulo de contagem.

Ao analisar os experimentos realizados na comunicação pode-se chegar a conclusão que os resultados obtidos foram bons. Tiveram perdas de pacotes e um certo atraso, porém nada que vá comprometer o sistema. As perdas de pacotes assim como o atraso fim-a-fim, ocorreram devido a mudança constante de posição do ônibus. Em determinados trechos a comunicação sem fio sofre com interferências e nesses pontos ocorre uma queda de comunicação ou reduz-se o sinal de transmissão. Para garantir que ao final do percurso os dados sejam íntegros e nenhuma contagem tenha se perdido, foi implementado um mecanismo de redundância.

No final do percurso, uma única mensagem contendo um resumo de todas as paradas e contagem é enviada para a empresa, visando assegurar a correção dos dados enviados anteriormente, tanto na comunicação interna, quanto externa. Os trabalhos analisados da literatura de um modo geral não se preocupam com a comunicação, apenas efetuam a contagem. Outros não tratam os dados embarcados, mas sim em computadores externos ao ônibus. Para isso os vídeos são coletados e transportados para um computador externo a fim de realizar os experimentos com a contagem. Conforme mencionado acima (BOREIKO; TESLYUK, 2016) transmite os vídeos para uma rede externa através de um modem 3G.

Se os trabalhos não levam em consideração a comunicação, tampouco são considerados os conceitos de Internet das Coisas e a agregação dos dados. A agregação e o tratamento dos dados embarcado no ônibus são itens importantes a serem considerados, visto que envolvem diversos fatores a facilitar o trabalho das empresas. Com o tratamento dos dados no ônibus, automaticamente o tamanho dos dados a serem transmitidos diminui, o que evita a necessidade de dispendir altos custos com transmissão ou até grandes estruturas. Mas para que o tratamento seja possível em cada porta e as informações sejam íntegras para transmitir para uma rede externa ao ônibus a comunicação interna entre dispositivos se mostra imprescindível. Então

vê-se a necessidade da implementação dos conceitos de Internet das Coisas.

Os resultados confirmaram que a arquitetura desenvolvida é adequada para o cenário alvo, e escalável para suportar diferentes tipos de hardware e protocolos de IoT. Porém alguns fatores podem ser melhorados, o que será abordado na próxima seção.

6.1 TRABALHOS FUTUROS

Este trabalho de dissertação abordou algumas implementações, como uma arquitetura, contagem de passageiros, agregação e comunicação. Na contagem foram utilizadas técnicas de visão computacional, enquanto na agregação e comunicação utilizou-se mecanismos de Internet das Coisas. Apesar da Arquitetura se mostrar eficaz, os itens que compõem ela são passíveis de melhorias.

A contagem não apresentou resultados ótimos devido a proximidade dos passageiros com a câmera. Analisou-se que os melhores resultados da literatura são com a imagem mais distante. Dessa forma, um dos trabalhos futuros é realizar experimentos com câmeras dotadas de um sensor de profundidade, como kinects por exemplo.

A técnica que apresentou melhor resultado foi YOLOV3, porém suas implementações apresentaram um alto tempo de demora no processamento, devido a pouco poder computacional do Raspberry. Dessa forma uma possível solução seria realizar experimentos com um microcontrolador com maior poder computacional e dotado de placas de processamento gráfico.

Na comunicação, os resultados obtidos foram bons, porém, podem ser obtidos resultados melhores com a implementação de protocolos mais leves como 6LoWPAN por exemplo. Uma forma de obter melhores resultados é desenvolver protocolos próprios baseados em protocolos de IoT, como CoAP por exemplo, a fim de diminuir o tempo de comunicação e processamento de agregação.

O trabalho aqui exposto, apesar de realizar todo o processo de tratamento, agregação e comunicação, inclusive com a rede externa, não contemplou o desenvolvimento da camada de aplicação da arquitetura. Assim um dos trabalhos futuros é desenvolver a camada de aplicação que pode ser composta por diversos componentes. Primeiramente um web server capaz de coletar os dados oriundos dos ônibus e armazená-los em um banco de dados para uso futuro. Após, a fim de tratar esses dados, serão necessários softwares para gestão das empresas assim como aplicativos para que os passageiros possam saber informações dos ônibus em tempo real.

REFERÊNCIAS

- AGARWAL, N.; AGARWAL, G. Role of cloud computing in development of smart city. **National Conference on Road Map for Smart Cities of Rajasthan (NC-RMSCR)**, p. 228–232, 2017.
- AKYILDIZ, I. F. et al. Wireless sensor networks: a survey. **Computer networks**, Elsevier, v. 38, n. 4, p. 393–422, 2002.
- ALBINO, V.; BERARDI, U.; DANGELICO, R. M. Smart cities: Definitions, dimensions, performance, and initiatives. **Journal of Urban Technology**, Taylor & Francis, v. 22, n. 1, p. 3–21, 2015.
- ARASTEH, H. et al. IoT-based smart cities: a survey. In: IEEE. **Environment and Electrical Engineering (EEEIC), 2016 IEEE 16th International Conference on**. [S.l.], 2016. p. 1–6.
- ASHTON, K. That ‘internet of things’ thing. **RFID Journal**, v. 22, n. 7, 2011.
- ATZORI, L.; IERA, A.; MORABITO, G. The internet of things: A survey. **Computer networks**, Elsevier, v. 54, n. 15, p. 2787–2805, 2010.
- BAKICI, T.; ALMIRALL, E.; WAREHAM, J. A smart city initiative: the case of barcelona. **Journal of the Knowledge Economy**, Springer, v. 4, n. 2, p. 135–148, 2013.
- BEGG DANIEL T.H. LAI, M. P. R. **Computational Intelligence in Biomedical Engineering**. [S.l.]: CRC Press, 2007.
- BEHNEL, S.; FIEGE, L.; MUHL, G. On quality-of-service and publish-subscribe. In: IEEE. **Distributed Computing Systems Workshops, 2006. ICDCS Workshops 2006. 26th IEEE International Conference on**. [S.l.], 2006. p. 20–20.
- BENEVOLO, C.; DAMERI, R. P.; D’AURIA, B. Smart mobility in smart city: Action taxonomy. **ICT Intensity and Public Benefits: Switzerland: Springer International Publishing [doi> 10.1007/978-3-319-23784-8_2]**, 2016.
- BERNINI, N. et al. An embedded system for counting passengers in public transportation vehicles. In: IEEE. **Mechatronic and Embedded Systems and Applications (MESA), 2014 IEEE/ASME 10th International Conference on**. Senigallia, Italy, 2014. p. 1–6.
- BOREIKO, O.; TESLYUK, V. Structural model of passenger counting and public transport tracking system of smart city. In: IEEE. **Perspective Technologies and Methods in MEMS Design (MEMSTECH), 2016 XII International Conference on**. Lviv, Ukraine, 2016. p. 124–126.
- CARVALHO, C. H. R. d. **Desafios da mobilidade urbana no Brasil**. [S.l.]: Instituto de Pesquisa Econômica Aplicada (Ipea), 2017. <<http://repositorio.ipea.gov.br/handle/11058/6664>>, Acessado em: Dezembro de 2017.

CHUN, B.-T.; LEE, S.-H. Review on its in smart city. **Advanced Science and Technology Letters**, v. 98, p. 52–54, 2015.

CORTES, C.; VAPNIK, V. Support-vector networks. **Machine Learning**, v. 20, n. 3, p. 273–297, Setembro 1995.

CRISTIANINI, N.; SHAW-ETAYLOR, J. **An introduction to support vector machines**. [S.l.]: Cambridge University Press Cambridge, United Kingdom, 2000.

DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: IEEE. **Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on**. San Diego, CA, USA, USA, 2005. v. 1, p. 886–893.

ESCOLANO, C. O. et al. Passenger demand forecast using optical flow passenger counting system for bus dispatch scheduling. In: IEEE. **Region 10 Conference (TEN-CON), 2016 IEEE**. Singapore, Singapore, 2016. p. 1875–1878.

FINKELZELLER, K. **The RFID handbook**. [S.l.]: John Wiley & Sons, 2003.

FREITAS, S. J. C. de; PLACIDES, F. M.; ARAÚJO, D. Os desafios da mobilidade urbana. **Percursos Acadêmicos**, p. 239–241, 2015.

FREUND, Y.; SCHAPIRE, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. **Journal of computer and system sciences**, Elsevier, v. 55, n. 1, p. 119–139, 1997.

GIFFINGER, R.; GUDRUN, H. Smart cities ranking: an effective instrument for the positioning of the cities? **ACE: Architecture, City and Environment**, Centre de Política del Sòl i Valoracions-Universitat Politècnica de Catalunya, v. 4, n. 12, p. 7–26, 2010.

GOMIDE, A. d. Á. **Mobilidade urbana, iniquidade e políticas sociais**. [S.l.]: Instituto de Pesquisa Econômica Aplicada (Ipea), 2006. <http://repositorio.ipea.gov.br/bitstream/11058/4511/1/bps_n.12_ensaio5_alexandre12.pdf>, Acessado em: Dezembro de 2017.

HAAR, A. Zur theorie der orthogonalen funktionensysteme. **Mathematische Annalen**, Springer, v. 69, n. 3, p. 331–371, 1910.

HALL, P. Creative cities and economic development. **Urban studies**, Sage Publications Sage UK: London, England, v. 37, n. 4, p. 639–649, 2000.

HARRISON, C. et al. Foundations for smarter cities. **IBM Journal of Research and Development**, IBM, v. 54, n. 4, p. 1–16, 2010.

HORN, B. K.; SCHUNCK, B. G. Determining optical flow. **Artificial intelligence**, Elsevier, v. 17, n. 1-3, p. 185–203, 1981.

KON, F.; SANTANA, E. F. Z. Cidades inteligentes: Conceitos, plataformas e desafios. **Jornadas de Atualização em Informática (JAI)**, Sociedade Brasileira de Computação, São Paulo, São Paulo, p. 48, 2016.

KUMAR, V.; TIWARI, S. Routing in ipv6 over low-power wireless personal area networks (6lowpan): A survey. **Journal of Computer Networks and Communications**, Hindawi, v. 2012, 2012.

KURIAN, M. Various object recognition techniques for computer vision. **Journal of Analysis and Computation**, v. 7, n. 1, p. 39–47, 2011.

KUSHALNAGAR, N.; MONTENEGRO, G.; SCHUMACHER, C. P. P. **IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals**. [S.l.], 2007. Disponível em: <<https://tools.ietf.org/pdf/rfc4919.pdf>>.

LENZ, B.; HEINRICHS, D. What can we learn from smart urban mobility technologies? **IEEE Pervasive Computing**, IEEE, v. 16, n. 2, p. 84–86, 2017.

LI, F. et al. Automatic passenger counting system for bus based on rgb-d video. In: **2nd Annual International Conference on Electronics, Electrical Engineering and Information Science (EEEIS 2016)**. [S.l.: s.n.], 2016. p. 209–220.

LIN, T.-Y. et al. Focal loss for dense object detection. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, 2018.

LOGITECH.COM. **Logitech**. 2017. <www.logitech.com/pt-br/product/hd-webcam-c270?crid=34#specification-tabular>, Acessado em: Dezembro de 2017. Disponível em: <<https://www.logitech.com/pt-br/product/hd-webcam-c270?crid=34#specification-tabular>>.

MICHALSKI, R. S.; CARBONELL, J. G.; MITCHELL, T. M. **Machine learning: An artificial intelligence approach**. [S.l.]: Springer Science & Business Media, 2013.

MOESLUND, T. B.; HILTON, A.; KRÜGER, V. A survey of advances in vision-based human motion capture and analysis. **Computer vision and image understanding**, Elsevier, v. 104, n. 2-3, p. 90–126, 2006.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. **Sistemas Inteligentes-Fundamentos e Aplicações**, v. 1, n. 1, p. 32, 2003.

MUKHERJEE, S. et al. Anovel framework for automatic passenger counting. In: IEEE. **Image Processing (ICIP), 2011 18th IEEE International Conference on**. Brussels, Belgium, 2011. p. 2969–2972.

NALEPA, J.; SZYMANEK, J.; KAWULOK, M. Real-time people counting from depth images. In: SPRINGER. **International Conference: Beyond Databases, Architectures and Structures**. [S.l.], 2015. p. 387–397.

NAMBIAR, R.; SHROFF, R.; HANDY, S. Smart cities: Challenges and opportunities. In: IEEE. **Communication Systems & Networks (COMSNETS), 2018 10th International Conference on**. [S.l.], 2018. p. 243–250.

NATIONS, U. **Urban and Rural Areas 2014**. 2014. <https://esa.un.org/unpd/wup/wallcharts/WUP_2014%20Urban-Rural%20Areas%20Wallchart.pdf>, Acessado em: agosto de 2017.

NODEMCU. **Integrated Development Environment (IDE)**. 2017. <http://www.nodemcu.com/index_en.html>, Acessado em: Dezembro de 2017. Disponível em: <http://www.nodemcu.com/index_en.html>.

PÁDUA, F. L.; CAMPOS, M. F.; CARCERONI, R. L. Real-time pedestrian counting based on computer vision. In: **Proc. of the Brazilian Symposium on Intelligent Automation**. Bauru, São Paulo: [s.n.], 2003. p. 1–6.

PERNG, J. W. et al. The design and implementation of a vision-based people counting system in buses. In: **2016 International Conference on System Science and Engineering (ICSSE)**. [S.l.: s.n.], 2016. p. 1–3.

POPPE, R. A survey on vision-based human action recognition. **Image and vision computing**, Elsevier, v. 28, n. 6, p. 976–990, 2010.

RASPBERRYPI.ORG. **Raspberry Pi 3 Model B**. 2017. <<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>>, Acessado em: Dezembro de 2017. Disponível em: <<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>>.

REDMON, J. et al. You only look once: Unified, real-time object detection. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 779–788.

REDMON, J.; FARHADI, A. YOLOv3: An incremental improvement. **arXiv preprint arXiv:1804.02767**, 2018.

SCHAPIRE, R. E. The strength of weak learnability. **Machine learning**, Springer, v. 5, n. 2, p. 197–227, 1990.

SCHMIDHUBER, J. Deep learning in neural networks: An overview. **Neural networks**, Elsevier, v. 61, p. 85–117, 2015.

SEPUD, S. D. P. U. E. D. S. **Joinville Cidade em Dados 2017**. 2017. <<https://www.joinville.sc.gov.br/wp-content/uploads/2016/01/Joinville-Cidade-em-Dados-2017.pdf>>, Acessado em: Dezembro de 2017.

SETHI, P.; SARANGI, S. R. Internet of things: architectures, protocols, and applications. **Journal of Electrical and Computer Engineering**, Hindawi, v. 2017, 2017.

SHELBY, Z.; HARTKE, K.; BORMANN, C. The constrained application protocol (coap). 2014.

SHENG, Z. et al. A survey on the ietf protocol suite for the internet of things: Standards, challenges, and opportunities. **IEEE Wireless Communications**, IEEE, v. 20, n. 6, p. 91–98, 2013.

SZELISKI, R. Computer vision: Algorithms and applications. In: SPRINGER. **Computer Vision: Algorithms and Applications**. [S.l.], 2010.

TIAN, D. ping et al. A review on image feature extraction and representation techniques. **International Journal of Multimedia and Ubiquitous Engineering**, Citeseer, v. 8, n. 4, p. 385–396, 2013.

VIANNA, G. S. B. Mobilidade urbana no brasil: uma estimativa do produto perdido em trânsito. **Monografia (Graduação em Ciências Econômicas)**. Instituto de Economia, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2013.

VILLAVARDE, B. C. et al. Constrained application protocol for low power embedded networks: A survey. In: IEEE. **Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on**. [S.l.], 2012. p. 702–707.

VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. In: IEEE. **Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on**. Kauai, HI, USA, USA, 2001. v. 1, p. I–I.

WIRESHARK.ORG. **Wireshark**. 2017. <<https://www.wireshark.org/>>, Acessado em: Dezembro de 2017. Disponível em: <<https://www.wireshark.org/>>.

YAHIAOUI, T.; KHOUDOUR, L.; MEURIE, C. Real-time passenger counting in buses using dense stereovision. **Journal of Electronic Imaging**, International Society for Optics and Photonics, v. 19, n. 3, p. 031202, 2010.

YAN-YAN, C. et al. Pedestrian detection and tracking for counting applications in metro station. **Discrete dynamics in nature and society**, Hindawi, v. 2014, 2014.

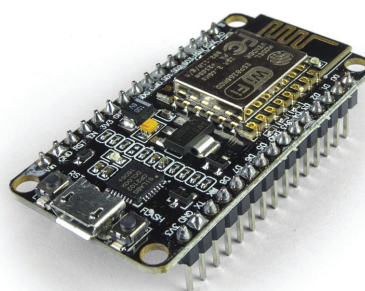
YUEN, H. et al. Comparative study of hough transform methods for circle finding. **Image and vision computing**, Elsevier, v. 8, n. 1, p. 71–77, 1990.

ANEXO A – REQUISITOS DE IMPLEMENTAÇÃO.

Com o objetivo de avaliar a abordagem proposta, experimentos foram realizados em ambiente real (ônibus em movimento). Na implementação utilizou-se os seguintes componentes de hardware conforme Figura A.1: Raspberry Pi versão 3b+, que possui um processador BCM2837 de 4 núcleos, 64 bits ARM Cortex-A53 a 1.2GHz, memória RAM de 1GB e um módulo de comunicação sem fio no padrão 802.11 b/g/n (RASPBERYPYPI.ORG, 2017); placa NodeMCU ESP32 (NODEMCU, 2017) com um módulo GPRS (A7 AI-Thinker); e uma câmera monocular da marca Logitech modelo HD Webcam C270 com uma resolução de 1280 x 720 pixels (LOGITECH.COM, 2017). O desenvolvimento do algoritmo para a contagem de passageiros (ver Capítulo 4) foi



(a) Raspberry Pi 3B+ - Módulo de Contagem



(b) NodeMCU ESP32 - Módulo de Agregação

Figura A.1 – (a) Hardware do Módulo de Contagem e (b) Módulo de Agregação

realizado na linguagem de programação Python versão 3.0 com auxílio da biblioteca de visão computacional Opencv na versão 3.3.0. Como ambiente de desenvolvimento para escrever o código de operação do NodeMCU, utilizou-se o Ambiente de Desenvolvimento Integrado (IDE) do Arduino na versão 1.8.1 devido a maior oferta de bibliotecas e à compatibilidade com a linguagem C/C++.