

UNIVERSIDADE FEDERAL DO PAMPA

Wesley Ferreira de Ferreira

**Um Sistema para Organização de Acervos
Baseado em Visão Computacional**

Alegrete
2020

Wesley Ferreira de Ferreira

Um Sistema para Organização de Acervos Baseado em Visão Computacional

Trabalho de Conclusão de Curso apresentado
ao Curso de Graduação em Ciência da Com-
putação da Universidade Federal do Pampa
como requisito parcial para a obtenção do tí-
tulo de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Marcelo Resende Thi-
elo

Alegrete
2020

Wesley Ferreira de Ferreira

Um Sistema para Organização de Acervos Baseado em Visão Computacional

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Trabalho de Conclusão de Curso defendido e aprovado em 11 de dezembro de 2020

Banca examinadora:



Prof. Dr. Marcelo Resende Thielo

Orientador
UNIPAMPA



Prof.ª Dra. Aline Vieira de Mello

UNIPAMPA



Prof. Me. Eliezer Soares Flores

UNIPAMPA

Este trabalho é dedicado à minha mãe que dedica a sua vida a preparar as crianças dos outros para o mundo, assim como fez comigo.

AGRADECIMENTOS

Agradeço, primeiramente à minha família, sem a qual não haveria suporte para a realização deste trabalho. Em especial, agradeço a minha mãe, Andréia Ferreira, pelo apoio emocional, financeiro e na revisão desta monografia. Agradeço aos meus amigos e amigas, Débora Siqueira, Paula Maisa Pereira, Felipe Bedinoto, Lucas Antunes, Sandy Andrade, Carina Maccari, dentre muitos outros, que direta ou indiretamente ajudaram no desenvolvimento deste trabalho. Agradeço aos meus colegas de equipe da Bode, que não deixaram de acreditar e confiar em meu trabalho, mesmo após momentos de ineficácia nas tarefas realizadas. Agradeço imensamente à minha terapeuta, Ediane Cappa, por me manter mental e fisicamente saudável, apesar do estresse e ansiedade gerados por um trabalho de conclusão de curso, bem como, por cada palavra de apoio e “puxão de orelha”, necessários neste momento. Agradeço ao meu orientador, Prof. Marcelo Resende Thielo, por propor a ideia deste trabalho, por aguentar minhas visitas e mensagens inoportunas e sem aviso prévio, por aguentar as minhas neuras, pelas discussões durante o desenvolvimento do mesmo, necessárias para a produção do mesmo, bem como pela atenção na revisão desta monografia. Agradeço também à minha banca examinadora Aline Vieira de Mello e Eliezer Soares Flores que fizeram valiosas contribuições para o presente trabalho. Agradeço a todos os colegas, por juntos termos superado essa importante etapa da vida. Por último, mas não menos importante, agradeço a Deus pela oportunidade de estar, aqui e agora, realizando o que sonhei e superando muitas provações.

“All we have to decide is what to do with the time that is given us.”
(J.R.R. Tolkien, *The Fellowship of the Ring*, 1954)

RESUMO

Mecanismos de busca são fundamentais com o aumento massivo de informação que é gerada todos os dias pela internet. De acordo com Cisco (2018), 82% destes dados estão no formato de imagens e vídeos. Portanto, é necessário o desenvolvimento de mecanismos robustos para indexação desse tipo de informação e também de extração de características e busca desta informação visual. Individualmente, cada um tem suas necessidades pessoais de produção de imagens e vídeos, a depender do estilo de vida, mas muitas delas têm um *smartphone* na mão. Para pessoas com vidas agitadas, que têm problemas de memória ou muito para lembrar, público alvo do trabalho, o fato de ter uma ferramenta dessas disponível pode ser de grande utilidade para recordar o que é importante, como a data de uma reunião ou onde deixou as chaves do carro, por exemplo. Os objetivos deste trabalho foram o desenvolvimento e a implementação de um protótipo de uma aplicação de organização de acervo pessoal em ambientes internos usando visão computacional e ferramentas auxiliares de processamento de texto e voz. O uso da ferramenta foi pensada para que seja possível, de forma rápida e prática, a organização dos pertences pessoais, consistindo ou não de uma coleção. Em uma investigação na literatura, foram encontrados diversos métodos e ferramentas que poderiam auxiliar viabilidade do trabalho, portanto, muitos desses tiveram que ser investigados e testados durante o desenvolvimento do protótipo. Dentre os métodos escolhidos para a composição do protótipo estão: o extrator de características ORB, a estrutura de árvore de vocabulário para armazenamento dos descritores e o Google *Speech-To-Text* para o processamento de voz. A validação foi realizada com uma grande quantidade de combinações de casos de teste para que se pudesse ter a noção das capacidades e imperfeições dos métodos escolhidos e da ferramenta que foi desenvolvida. Dentre elas o que se destaca é a capacidade de aceitar diversas inserções consecutivas sem a necessidade de retreinamento a cada inserção, necessitando, no entanto, de um reagrupamento dos descritores da árvore, para que o tempo de processamento não cresça e o nível de qualidade alcançado, de mais de 80%, se mantenha em um ambiente com um imenso número de variáveis, como: foco da câmera, desfoque de movimento, iluminação, sombra, etc. Como conclusão deste trabalho, o qual, apesar de possuir ainda muitos pontos de melhoria, temos que o objetivo proposto foi cumprido e que, com as devidas melhorias, o protótipo desenvolvido viabiliza um produto real capaz de gerar um impacto positivo na vida do público alvo.

Palavras-chave: Visão Computacional. Organizador de Acervos. Imagem. Áudio. Texto.

ABSTRACT

Search engines have become essential with the massive increase in the amount of information generated everyday in the internet. According to Cisco (2019) 82% of these information consists of images and videos. Therefore the development of robust mechanisms to index this type of information, extract characteristics and search this visual information is necessary. Individually, each person has their personal needs for producing images and videos, depending on their lifestyle, but many have a smartphone in hand. For people with busy lives, who have memory problems or many things to remember, having a tool quickly available can be very useful in remembering things, like the date of a meeting or where you left your car keys. The objectives of this work are the development and implementation of a prototype of an application for the indoor organization of personal collections using computer vision and auxiliary tools for text and voice processing. The application was designed to be an easy and practical tool for using to organize personal belongings, whether or not consisting of a collection. In a literature search for a starting point to the design of this prototype, several methods and tools that could assist in its feasibility were found, however, no work with a similar proposal was identified. Therefore, many of the methods in this work had to be investigated, implemented and tested during the development of the prototype. The methods selected for the composition of the prototype are: ORB feature extractor, the vocabulary tree structure for storing descriptors and Google Speech-To-Text for speech processing. The validation was carried out with a large number of combinations of test cases so that one could have an idea of the capabilities and imperfections of the tool that was developed. Among them, what stands out is the ability to accept several consecutive insertions without the need for retraining at each insertion, however needing a regrouping of the tree descriptors, so that the processing time does not increase and the level of quality achieved, of more than 80 %, remain in an environment with an immense number of variables, such as: camera focus, motion blur, lighting, shadow, etc. As a conclusion of this work, which, although it still has many points for improvement, we have that the proposed objective was fulfilled and that, given the required improvements, the prototype developed could be turned into a real product capable of generating a positive impact on the life of the target audience.

Key-words: Computer Vision. Collection Organizer. Image. Audio. Text.

LISTA DE FIGURAS

Figura 1 – Ilustração do vetor de descritores de características.	29
Figura 2 – Ilustração dos arquivos invertidos.	33
Figura 3 – Ilustração da árvore de vocabulário.	34
Figura 4 – Diagrama de pacotes do cliente.	44
Figura 5 – Diagrama de pacotes do servidor.	45
Figura 6 – Diagrama de fluxo da aplicação:	46
Figura 7 – Captura da tela do servidor	49
Figura 8 – Ilustração do algoritmo de detecção de movimentação e desaparecimento.	49
Figura 9 – Livro centralizado, a esquerda, e em detalhes, com os <i>keypoints</i> do FAST orientado e BRIEF rotacionado, do inglês <i>Oriented FAST and Rotated BRIEF</i> (ORB), na direita.	50
Figura 10 – Caso de teste considerado excelente.	52
Figura 11 – Caso de teste considerado aceitável.	52
Figura 12 – Caso de teste considerado insatisfatório.	53
Figura 13 – Objetos utilizados nos testes.	53
Figura 14 – Quantidade de sucessos e fracassos para os duzentos testes realizados.	54
Figura 15 – Tempo em segundos de cada uma das inserções.	55
Figura 16 – Porcentagem de cada um dos casos de categorização.	55
Figura 17 – Tempos em segundos da consulta textual.	56
Figura 18 – Tempo da consulta visual do teste manual.	57
Figura 19 – Tempo em segundos de inserção por vídeo.	57
Figura 20 – Tempo em segundos de inserção por quadro.	58
Figura 21 – Tempo da consulta visual do teste automatizado.	58

LISTA DE TABELAS

Tabela 1 – Trabalhos Relacionados	41
Tabela 2 – Tabela dos tempos em nanosegundos das dez consultas visuais de cada objeto	56
Tabela 3 – Tabela dos tempos das dez inserções de cada objeto na gaveta	71
Tabela 4 – Tabela dos tempos das dez inserções de cada objeto na prateleira	72
Tabela 5 – Tabela dos tempos das dez inserções de cada objeto no roupeiro	72
Tabela 6 – Tabela dos tempos das dez inserções de cada objeto na caixa	73
Tabela 7 – Tabela das qualidades e tempos das dez consultas de cada objeto na gaveta	75
Tabela 8 – Tabela das qualidades e tempos das dez consultas de cada objeto na prateleira	76
Tabela 9 – Tabela das qualidades e tempos das dez consultas de cada objeto no roupeiro	76
Tabela 10 – Tabela das qualidades e tempos das dez consultas de cada objeto na caixa	77

LISTA DE SIGLAS

- BoW** Conjunto de palavras, do inglês *Bag of Words*
- BRISK** Pontos chave escalável, invariante, binário e robusto, do inglês *Binary Robust Invariant Scalable Keypoints*
- FAST** Características do segmento de teste acelerado, do inglês *Features from Accelerated Segment Test*
- IA** Inteligência Artificial
- IOT** Internet das coisas, do inglês *Internet of Things*
- MSER** Regiões extremas maximamente estáveis, do inglês *Maximally Stable Extremal Regions*
- ORB** FAST orientado e BRIEF rotacionado, do inglês *Oriented FAST and Rotated BRIEF*
- RANSAC** Consenso de Amostra Aleatória, do inglês *Random Sample Consensus*
- SIFT** Transformação de características invariante de escala, do inglês *Scale Invariant Feature Transform*
- SURF** Características robustas aceleradas, do inglês *Speeded-Up Robust Features*
- VC** Visão Computacional

SUMÁRIO

1	INTRODUÇÃO	23
1.1	Motivação	23
1.2	Objetivos	24
1.3	Organização deste trabalho	25
2	FUNDAMENTAÇÃO TEÓRICA	27
2.1	Visão computacional	27
2.2	Reconhecimento de padrões em imagens	28
2.2.1	Extração de características	28
2.3	OpenCV	29
2.3.1	Principais módulos da OpenCV	30
2.4	Python	30
2.5	Agrupamento de dados	31
2.5.1	Métodos de agrupamento	31
2.5.2	Agrupamento e correspondência de imagens	32
2.6	Árvore de vocabulário	33
2.7	Processamento de fala e linguagem	34
3	TRABALHOS RELACIONADOS	37
3.1	Protocolo de pesquisa	37
3.2	Resultados da pesquisa	38
3.3	Considerações do capítulo	40
4	MATERIAIS E MÉTODOS	43
4.1	Arquitetura da aplicação	43
4.1.1	Aspectos do cliente	43
4.1.2	Aspectos do servidor	43
4.2	Infraestrutura da aplicação com relação ao servidor	44
4.3	Infraestrutura da aplicação com relação ao cliente	45
4.4	Ambiente de execução da aplicação	47
4.5	Processo de desenvolvimento	47
4.6	Protótipo desenvolvido	48
5	RESULTADOS E DISCUSSÕES	51
5.1	Procedimento de validação	51
5.2	Resultados obtidos	54
5.2.1	Resultados da etapa 1	54
5.2.2	Resultados da etapa 2	54
5.2.3	Resultados da etapa 3	55

5.2.4	Resultados da etapa 4	56
5.3	Análise dos resultados	58
6	CONCLUSÃO	61
	REFERÊNCIAS	63
	APÊNDICES	69
	APÊNDICE A – TEMPOS E SUCESSOS DAS INSERÇÕES	71
	APÊNDICE B – TEMPOS E QUALIDADES DAS CONSULTAS TEXTUAIS	75

1 INTRODUÇÃO

O uso de mecanismos de busca facilmente acessíveis, por meio de computadores ou *smartphones*, tem avançado sobre onde outrora predominava o uso de livros e outros materiais de consulta físicos. Dessa forma, a pesquisa sobre tópicos específicos tem se tornado cada vez mais ágil. Esse fenômeno pode ser observado em diversos ambientes, quando muitas bibliotecas já não possuem mais livros de indexação das prateleiras, não se precisa mais ir às lojas físicas para achar o produto que se quer comprar, nem ir até uma locadora para escolher um filme. Atualmente, já existem *softwares* robustos para resolver os problemas de indexação desse tipo de informação, como os utilizados por grandes livrarias online, ao exemplo da Livraria Saraiva (SARAIVA E SICILIANO S.A). Grandes varejistas como *Amazon* (AMAZON.COM, INC.) e serviços de *streaming* de vídeo conhecidos: *Netflix* (NETFLIX INC.), *Prime Video* (AMAZON.COM, INC.), etc.

Com o aumento massivo de dispositivos conectados à internet, estimado em 3,6 dispositivos per capita até 2022, a quantidade de informação que pode ser obtida por meio de ferramentas de indexação e busca cresce na mesma proporção. Estima-se que até 2022 o tráfego de conteúdo em imagem/vídeo compreenda 82% de todo o tráfego da Internet (CISCO, 2019). Mecanismos de armazenamento e busca de vídeos e fotos têm que lidar com milhares de gigabytes de conteúdo gerenciado a mais por dia, como pode ser visto em tempo real no serviço *Internet Live Stat* (DADAX).

A Visão Computacional (VC), sub-área da Ciência da Computação, responsável pelo processamento e interpretação de imagens tentando simular o comportamento de extração e reconhecimento de informação da visão humana, em crescimento acelerado há mais de 20 anos, tem tido grandes avanços, no entanto, ainda há várias dificuldades que permeiam essa área, uma das principais é a de extrair contexto semântico de alto nível usando características visuais de baixo nível (ZHOU; LI; FELLOW, 2017).

Para uma indexação e busca eficiente por objetos em bancos de imagens, é necessário que haja mecanismos eficazes de extração de características destes objetos. Também é necessário que haja estratégias eficientes para organizar, comparar e recuperar as características obtidas de forma rápida, precisa e confiável.

No início da década de 2000, alguns foram os trabalhos notórios nesses segmentos. Dentre os pioneiros que se destacaram estão os métodos SIFT (LOWE, 2004) e BoW (SIVIC; ZISSERMAN, 2003), que abriram caminho para avanços significativos na área (ZHOU; LI; FELLOW, 2017).

1.1 Motivação

As síndromes demenciais, como a Síndrome de Alzheimer, são um problema crescente no Brasil e no mundo (GUTIERREZ et al., 2014). Pessoas com esse tipo de síndrome têm perda gradual da memória de curto prazo nos estágios iniciais da doença, perda da

memória de longo prazo e perda da capacidade de manter grandes linhas de raciocínio em estágios mais avançados. Para essas pessoas, lembrar onde seus pertences pessoais estão guardados não é uma tarefa trivial. Por isso, o uso de tecnologias emergentes, como a VC e a Inteligência Artificial (IA), pode ajudar essas pessoas em diversos aspectos de suas vidas, por exemplo, ajudar a localizar objetos esquecidos.

Em outra situação, por exemplo, coleções de itens são complexas de gerenciar por si só. Com o crescimento do número de objetos e variedades, o nível de complexidade cresce na mesma proporção. Gerenciar uma coleção com milhares de peças usando sistemas tradicionais, como um sistema de informação de catálogo simples ou até mesmo um livro é uma tarefa trabalhosa, demorada e enfadonha. Tecnologias de VC, bem como tecnologias sofisticadas de busca e recuperação de informação podem reduzir significativamente sua complexidade, de modo a torná-lo automatizado e preditivo, imprimindo inteligência na busca de objetos de interesse dentro de uma coleção.

Com o aumento de agitação da vida moderna, o estresse, a ansiedade e por consequência os problemas de memória, pela falta de atenção, são cada vez mais presentes na atualidade. Cury (2014) afirma que vivemos numa sociedade urgente, rápida, ansiosa e estressada como nunca antes. E, por este motivo, pessoas de todas as idades estão indo ao esgotamento mental, levando não só ao aumento do estresse como também ao aumento do deficit de concentração e ao deficit de memória. Essas pessoas, que possuem uma rotina atarefada e frequentemente sobrecarregada, muitas vezes expostas a problemas como o stress e a síndrome de *Burnout* (MASLACH; LEITER, 2016), acabam eventualmente negligenciando aspectos de sua vida pessoal, como a organização de seus pertences, largando-os de qualquer jeito ou então guardando-os sem prestar realmente atenção onde os deixaram. Mesmo para uma pessoa de rotina "normal", é natural ter alguma dificuldade em localizar seus objetos, sendo bem-vinda uma ferramenta para o auxílio na localização desses pertences da forma mais simples possível.

Visto o número de pessoas e casos onde um organizador pessoal rápido e eficiente pode ser de grande utilidade, a possibilidade de criação de uma aplicação de organização pessoal baseada em imagem contribuirá significativamente para a melhora na qualidade de vida dessas pessoas que poderão contar com uma ferramenta no bolso para localizar seus objetos sem maior esforço. Desse modo, propomos o desenvolvimento do protótipo para esse fim.

1.2 Objetivos

O objetivo geral deste trabalho é a construção de um mecanismo de indexação e busca, com base em imagem, texto e voz, para aplicação pessoal, a partir de bases de dados construídas pelos próprios usuários. Mais especificamente, foi investigada a implementação de um mecanismo voltado para a organização do acervo pessoal em ambientes internos, com o auxílio de sistemas de posicionamento baseados em *wireless fingerprint*.

Para isso, foram necessários estudos em diferentes áreas da ciência da computação, tais como: VC, Processamento de Voz, Redes de Computadores e Bancos de Dados. Os principais objetivos específicos e desafios que foram abordados por este trabalho são:

- Detecção e rastreamento da movimentação do objeto na cena.
- Extração de descritores de características visuais.
- Processamento de informações extras definidas pelo usuário, por texto ou voz.
- Armazenagem e recuperação das informações na estrutura definida e base de dados à medida que são manipuladas pelo sistema.
- Comunicação confiável entre o cliente e o servidor.

1.3 Organização deste trabalho

O restante deste trabalho está organizado da seguinte forma:

- **Capítulo 2 - Fundamentação Teórica:** são discutidos conceitos fundamentais para o entendimento do presente trabalho.
- **Capítulo 3 - Trabalhos Relacionados:** é apresentado o mapeamento sistemático realizado para investigar o estado da arte na área de estudo desse trabalho.
- **Capítulo 4 - Materiais e Métodos:** são descritos a arquitetura do protótipo implementado, os conjuntos de atividades realizadas para a construção deste, bem como as tecnologias aplicadas no mesmo.
- **Capítulo 5 - Resultados e Discussões:** são apresentados o procedimento de validação do protótipo, os testes realizados, os resultados obtidos e, por fim, a análise e discussão dos resultados.
- **Capítulo 6 - Conclusão:** São feitas as considerações sobre o trabalho realizado bem como os problemas enfrentados e possibilidades de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

O capítulo está organizado da seguinte forma: na seção 2.1 são introduzidos conceitos de VC; na seção 2.2 são apresentados conceitos básicos de reconhecimento de padrões; na seção 2.3 é introduzida uma biblioteca de código aberto para trabalhos de VC; na seção 2.4 é abordada em um panorama geral a linguagem de programação Python; na seção 2.5 são abordados conceitos básicos de agrupamentos e apresentados alguns métodos clássicos para este fim; na seção 2.6 é abordado o conceito por trás da classificação e recuperação de imagens a partir de árvores de vocabulário; por fim, na seção 2.7 são apresentados conceitos básicos sobre processamento de texto e voz.

2.1 Visão computacional

A VC é uma área interdisciplinar que envolve estudos profundos nas mais diversas áreas do conhecimento, dentre elas a Biologia, a Matemática, a Inteligência Artificial e a Robótica. Estas disciplinas são estudadas em conjunto para entender o conceito de imagem, o qual envolve uma série de aspectos de como o olho e o cérebro humano produzem e interpretam a informação visual, de modo a viabilizar o processamento da imagem digital adquirida por sensores, tornando-a tratável por computadores. Assim, pode-se tentar simular o comportamento humano de entendimento de uma imagem por uma máquina, a fim de extrair informações e tomar decisões com base nestas.

De acordo com Conci (2008), a área da VC trata da extração de informações, da identificação e classificação de objetos contidos em imagens. Para isso, envolve técnicas de IA ou de tomada de decisão. Os sistemas de VC vêm sendo usados em reconhecimento de pessoas, de assinaturas, de objetos; inspeção de peças em linhas de montagem; orientação de robôs em indústrias automatizadas, etc.

De acordo com Shapiro e Stockman (2001), o objetivo principal da VC é de tomar decisões sobre objetos e cenas reais com base em imagens capturadas. Para isso, é necessária a construção de descrições e modelos. Por este motivo, muitos especialistas dizem que VC tem por objetivo a construção de descrições e o reconhecimento de cenas presentes nas imagens.

De acordo com Krishna (2017), VC pode ser definida de duas formas: i) como uma área da ciência que estuda a extração e reconhecimento de informações de imagens digitais, ou ii) como a criação de algoritmos capazes de entender o conteúdo de imagens e usá-las para outras aplicações. A VC pode ser considerada parte da ciência da computação, onde as teorias de algoritmos e aprendizagem de máquina são essenciais para o seu desenvolvimento.

2.2 Reconhecimento de padrões em imagens

Reconhecimento de padrões é uma área de estudo fundamental para a VC, uma vez que, é por meio dela que as imagens são processadas e transformadas em informações que podem ser mais facilmente interpretadas por um computador. Para tal, várias abordagens são adotadas, como extração dos padrões de cor, extração dos padrões de textura, detecção de formas e bordas, dentre outras.

De acordo com Gonzalez (2009), padrão é a descrição quantitativa ou estrutural de um objeto ou entidade de interesse em uma imagem, em geral formado por um ou mais descritores. Uma classe de padrões é uma família de objetos que compartilham algumas características comuns. Ainda que, ligeiramente diferentes de alguma forma, como, por exemplo em termos da largura e do comprimento no caso de pétalas de flor, flores também irão variar, não apenas entre as diferentes classes de flores, mas também dentro de uma mesma classe (FISHER, 1936). O reconhecimento de padrões por uma máquina envolve técnicas para associação automática de padrões às suas respectivas classes de interesse. Na prática, os principais arranjos de padrões utilizados são vetores para descrições quantitativas e cadeias e árvores para descrições estruturais.

De acordo com Shapiro e Stockman (2001), uma classe é um conjunto de objetos com algumas propriedades em comum, a qual pode ser associada com um determinado rótulo. A classificação, por sua vez, é o processo de atribuir um rótulo para o objeto que deseja-se classificar, com base em uma representação das suas propriedades, sendo o classificador o dispositivo ou algoritmo responsável pelo processo de classificação. Cabe ainda mencionar que o procedimento de representar um objeto através das suas propriedades discriminativas é denominado extração de características, por exemplo, para classificar vegetais pode-se representá-los através das suas cores, formatos e texturas.

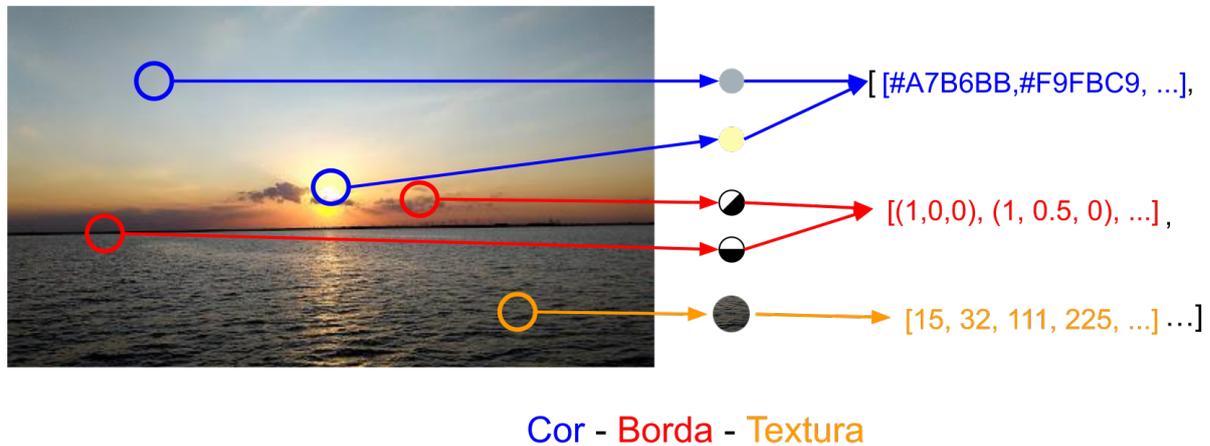
2.2.1 Extração de características

Correspondência entre imagens é um aspecto fundamental de muitos problemas na VC, incluindo geração de contexto tridimensional, reconhecimentos de objetos ou cenas e rastreamento de movimentos (LOWE, 2004). Descritores de características são valores discretos que expressam algumas propriedades de uma imagem tais como: formas, bordas, cores e texturas, como ilustrado na Figura 1.

De acordo com Szeliski (2011), extração de características é um componente essencial de muitas aplicações de VC. Dentre algumas das abordagens de detecção de características estão: i) a detecção de pontos de interesse, como picos de montanhas e quinas de construções. ii) a detecção de bordas, como a fronteira entre uma montanha e o céu. Um bom exemplo do processo de localização de características é apresentado no trabalho onde foi introduzido o algoritmo SIFT (LOWE, 2004). Outros métodos que também merecem destaque pois são amplamente citados na literatura a respeito são:

SURF (BAY; TUYTELAARS; GOOL, 2006), ORB (RUBLEE et al., 2011) e o BRISK (LEUTENEGGER; CHLI; SIEGWART, 2011).

Figura 1 – Ilustração do vetor de descritores de características.



Fonte: Proprio Autor

2.3 OpenCV

OpenCV (BRADSKI, 2000) é uma biblioteca gratuita de VC, de código aberto, escrita em C e C++ com desenvolvimento ativo para várias linguagens de programação como Python, Ruby, MatLab, dentre outras. Estando disponível para os principais sistemas operacionais do mercado (Windows, Linux, MacOS, Android, etc). A OpenCV dispõe de centenas de métodos de visão computacional e, adicionalmente, por causa de proximidade das áreas e o uso intrínseco no propósito base da OpenCV, uma biblioteca de aprendizagem de máquina de propósito geral. A OpenCV foi projetada para ser computacionalmente eficiente devido ao seu grande foco em aplicações de tempo real, tendo otimização para processamento paralelo e em interfaces Intel (BRADSKI; KAEHLER, 2008).

Segundo Bradski e Kaehler (2008), uma das principais metas da OpenCV é prover uma interface de fácil uso para a construção de aplicações em VC de forma livre, sendo a licença de código livre estruturada para que não haja nenhuma restrição para uso comercial. Em parte, devido aos termos de sua licença, essa biblioteca possui uma comunidade ampla e é utilizada por grandes companhias como HP, IBM, Microsoft, Intel e Google, bem como por alguns dos maiores centros de pesquisa como MIT, Stanford University, Cambridge University, etc.

Por ser uma biblioteca popular ao redor do mundo, possui grandes comunidades em países como China, Japão, Rússia, Europa e Israel. Dispõe também de uma comunidade

no *Yahoo* com quase 50 mil membros. É utilizada em diversas áreas, incluindo aplicações médicas, de satélite e mapas, veículos autônomos, dentre outros (BRADSKI; KAEHLER, 2008).

2.3.1 Principais módulos da OpenCV

A OpenCV tem uma estrutura modular, o que significa que o pacote inclui várias bibliotecas compartilhadas ou estáticas. Alguns dos módulos disponíveis são:

- **core** - um módulo compacto que define basicamente estruturas de dados, incluindo o vetor multi-dimensional denso “Mat” e funções básicas usadas em outros módulos;
- **imgproc** - um módulo de processamento de imagens que inclui filtragem não linear e linear de imagem, transformações geométricas de imagens (redimensionamento, distorção afim e de perspectiva, remapeamento genérico baseado em tabelas), conversão de cores, histogramas e mais;
- **video** - um módulo de análise de vídeo que inclui estimativa de movimento, subtração de fundo e algoritmos de rastreamento de objetos;
- **features2d** - um módulo que inclui algoritmos para detecção e descrição de características, tais como FAST, MSER, ORB e BRISK;

2.4 Python

O Python é uma linguagem de programação de propósito geral, multi-paradigma, dinamicamente tipada, e interpretada. Criada por Guido Van Rossum em 1991, é uma linguagem de *script* que tem como filosofia a simplicidade, clareza de código e produtividade (ROSSUM, 1995). Por ser uma linguagem interpretada, seu desempenho é inferior se comparado a linguagens compiladas como C e C++, podendo entretanto ser estendida com módulos nessas linguagens, por meio das quais bibliotecas com alta performance como o OpenCV são integradas a linguagem.

Além do OpenCV já apresentado, o Python, por ser uma linguagem fácil de usar e extensível, viabiliza que muitas bibliotecas sejam criadas por empresas e pela comunidade e distribuídas gratuitamente por plataforma independente ou pelo próprio indexador de pacotes da linguagem. Algumas das bibliotecas em destaque estão:

- **NumPy (HARRIS et al., 2020)**: biblioteca altamente otimizada para a operação de matrizes multi-dimensionais, álgebra linear, números randômicos e complexos. Baseada nas linguagens C++ e Fortran, é hoje, uma das bases das operações matemáticas do porte do OpenCV para Python.

- **scikit-learn (PEDREGOSA et al., 2011)**: biblioteca especializada em aprendizagem de máquina que também contém métodos de análise de dados, provê classes para classificação, regressão, *clustering*, normalização e extração de características;
- **Pandas (MCKINNEY, 2010)**: biblioteca especializada em análise de dados amplamente utilizada para mineração e pré-processamento de dados em passos anteriores às aplicações dos métodos de aprendizagem de máquina.
- **dlib (KING, 2009)**: biblioteca escrita em C++ portada para Python com ferramentas e métodos de aprendizado de máquina para a criação de softwares complexos para a solução de problemas reais.

2.5 Agrupamento de dados

Segundo Han, Kamber e Pei (2012), análise de grupos, ou simplesmente agrupamento (*clustering*) é o processo de particionamento de um conjunto de dados de objetos ou observações em subconjuntos. Cada subconjunto é um grupo dentro do qual cada objeto é similar a outro, e diferente de objetos em outro grupo. A análise de grupos de forma algorítmica é útil e pode ajudar a identificar grupos antes desconhecidos nos dados.

Ainda segundo Han, Kamber e Pei (2012), técnicas de agrupamento são usadas em muitas aplicações como, por exemplo, em inteligência empresarial, reconhecimento de padrões em imagens, busca na internet, biologia e segurança. Em reconhecimento de imagens, o agrupamento pode ser usado, por exemplo, para reconhecer caracteres escritos a mão.

Alpaydin (2009) afirma que em muitas aplicações, a amostra não pertence a apenas um grupo, mas vários, como reconhecimento de caracteres escritos a mão, que pode variar dependendo da região, bem como no reconhecimento de fala, onde a mesma palavra pode ser falada de formas variadas dependendo da pronúncia, sotaque, gênero, idade e outros. Pode-se caracterizar, por exemplo, com uma mesma amostra, o gênero e a idade de uma pessoa.

De acordo com Han, Kamber e Pei (2012), é difícil apresentar todos os métodos de agrupamento, pois, estes são muitos. No entanto, é importante que seja visto um panorama geral dessas abordagens, frequentemente dividido em métodos de particionamento, hierárquicos, baseados em densidade e baseados em grades. Alguns dos métodos abordados por Han, Kamber e Pei (2012) podem ser vistos como casos particulares das estratégias tratadas no decorrer desta seção.

2.5.1 Métodos de agrupamento

Boa parte dos métodos das medidas de similaridade/dissimilaridade são baseadas em distâncias (PADILHA; CARVALHO, 2017). Dado um número k , o número de

partições a serem identificadas, é criado um particionamento inicial, o qual é melhorado iterativamente movendo objetos entre os grupos. A estratégia de agrupamento hierárquico funciona organizando os grupos hierarquicamente em uma árvore. Métodos de agrupamento por densidade utilizam objetos na vizinhança para estipular uma densidade enquanto métodos baseados em grades usam grades multidimensionais que quantificam o espaço em um número finito de células. Uma breve descrição de alguns dos algoritmos mais clássicos, mais especificamente, *k-Means*, *k-Medoids* e agrupamento hierárquico probabilístico, podem ser vistos a seguir:

- **k-Means:** um dos algoritmos mais simples, agrupa os dados em k grupos iniciais de forma aleatória, e utiliza a distância euclidiana para o aprimoramento destes grupos.
- **k-Medoids:** organizado de forma que cada grupo possua um objeto representativo, também chamado de semente (*seed*). Assim se difere do *k-Means* que usa puramente a média dos objetos dentro de um grupo como referência, o que pode causar resultados discrepantes na presença de valores atípicos. O *k-Medoids* utiliza a soma das dissimilaridades de cada item e seu objeto representativo para minimizar a soma das diferenças entre cada objeto e a semente, considerando sempre a troca da semente por uma mais representativa.
- **Agrupamento Hierárquico Probabilístico:** Este método vem para corrigir uma série de desvantagens de outros métodos hierárquicos, tais como: problemas com dados parcialmente observados e métodos de cálculo de distância ineficientes. Para isso propõe-se o uso de modelos probabilísticos para a medição das distâncias entre os grupos.

2.5.2 Agrupamento e correspondência de imagens

A correspondência de imagens na VC, a partir de grupos criados, é fundamental em muitos aspectos, como reconhecimentos de objetos ou cenas e rastreamento de movimentos (LOWE, 2004). Para imagens, o principal artefato usado para o agrupamento e correspondência são os descritores extraídos por métodos como os descritos na subseção 2.2.1.

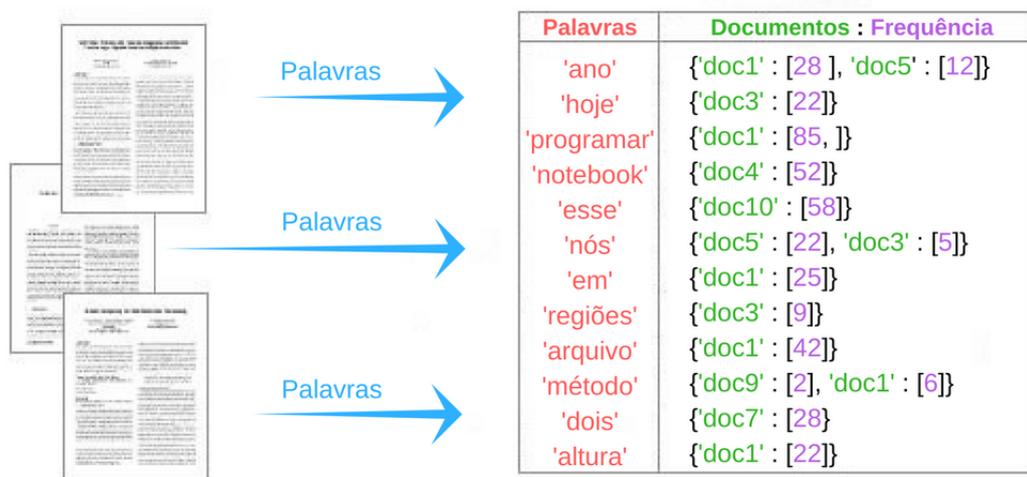
Conforme poderá ser visto no Capítulo 3, um número grande de métodos de categorização e correspondência de imagens são utilizados nos mais diversos tipos de aplicação. Dentre as diversas estruturas, *Bag of Words* (KABBAI; ABDELLAOUI; DOUIK, 2019) e Árvores de Vocabulário (LYRA, 2018) julgamos serem adequadas para compor a aplicação proposta neste trabalho.

2.6 Árvore de vocabulário

De acordo com Nister e Stewenius (2006), reconhecimento de objetos é um dos principais problemas na VC e é um tópico que tem sido extensivamente investigado. Um dos mais importantes desafios é construir métodos que se comportam bem com o crescimento do tamanho da base de dados, podendo selecionar um ou mais objetos em um tempo aceitável.

Em seus estudos IEEE (2003) apresentam uma abordagem de reconhecimento de objetos a partir de recuperação textual, mais especificamente com o uso de arquivos invertidos e algoritmos de agrupamento, representando as imagens em questão através de palavras visuais, ou seja, os descritores de características das imagens, como as palavras a serem reconhecidas em um vocabulário. Uma ilustração do funcionamento dos arquivos invertidos pode ser visto na Figura 2.

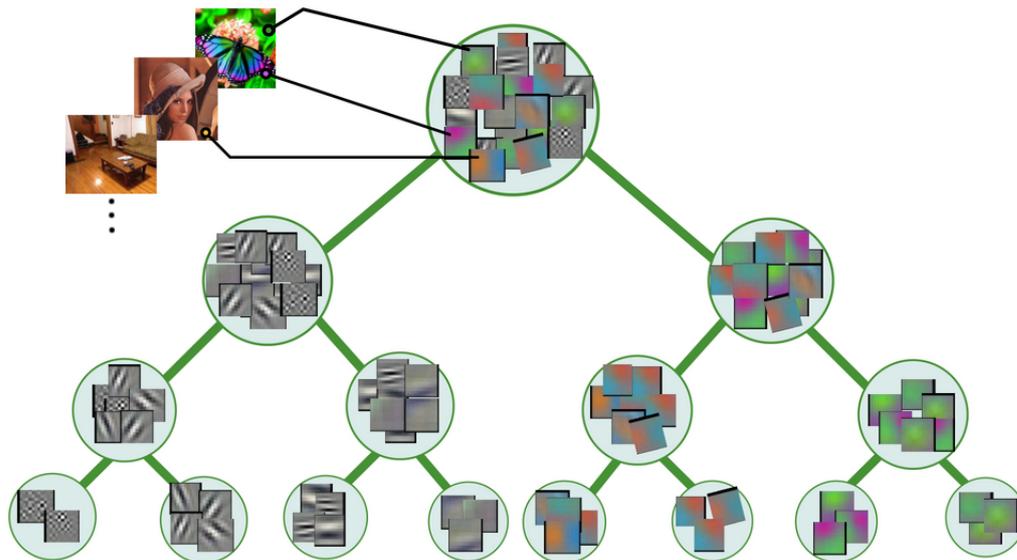
Figura 2 – Ilustração dos arquivos invertidos.



Fonte: Lyra (2018)

Inspirado no trabalho de IEEE (2003), Nister e Stewenius (2006) utiliza em seu trabalho a abordagem de palavras visuais e arquivos invertidos para propor um método de classificação e recuperação de imagens, baseado em uma estrutura de árvore, no qual os descritores são agrupados utilizando o algoritmo de *k-means* e utilizando a frequência dos termos dos arquivos invertidos como a relevância de uma imagem na consulta. Uma representação visual do funcionamento da árvore de vocabulário pode ser visto na Figura 3.

Figura 3 – Ilustração da árvore de vocabulário.



Fonte: Lyra (2018)

2.7 Processamento de fala e linguagem

O processamento da fala em particular e o processamento da linguagem natural em geral constituem, nos dias de hoje, o núcleo das tecnologias linguísticas e um dos novos paradigmas da língua e da linguística (BRAGA et al., 2008). Essas tecnologias permeiam nossa vida no dia de hoje, sob a forma de assistentes virtuais, leitores de livros digitais, sistemas controlados por voz, gps, sendo as possibilidades de aplicação quase ilimitadas.

A Linguística possui cem anos de história como uma disciplina científica, e a linguística computacional possui cinquenta como uma parte da ciência da computação. No entanto, foi apenas aproximadamente na última década que o entendimento de linguagens emergiu na indústria, alcançando milhões de pessoas, com recuperação de informação e a tradução por máquinas disponível na internet, bem como o reconhecimento de fala se tornando comum em computadores pessoais. Esta indústria se dispôs para avanços teóricos na representação e processamento de informações de linguagem. (JURAFSKY; MARTIN, 2008, Tradução Nossa)

De acordo com Jurafsky e Martin (2008), o que distingue o processamento de linguagem de outro sistema de processamento é o conhecimento da linguagem, e que sofisticados agentes de conversação, sistemas de tradução e sistemas de perguntas e respostas precisam ter um domínio amplo e profundo de conhecimento da linguagem.

Ainda de acordo com Jurafsky e Martin (2008), para muitos a habilidade de computadores de processar linguagens tão habilmente como nós, humanos, irá significar a chegada de máquinas realmente inteligentes. A base dessa crença é o fato de que o uso efetivo da linguagem está interligado com nossas habilidades cognitivas gerais.

De acordo com Russell e Norvig (2009), linguagens naturais são difíceis de lidar, não podem ser categorizadas como conjunto definido de sentenças. Uma linguagem natural é muito ampla, ambígua e está em constante mudança. Então, nossos modelos de linguagens são, na melhor das hipóteses, uma aproximação.

Ainda de acordo com Russell e Norvig (2009), reconhecimento de fala é a tarefa de identificar uma sequência de palavras proferidas por um falante, dado o sinal sonoro. Isso tem se tornado uma das aplicações principais da IA, milhões de pessoas interagem com sistemas de reconhecimento de fala todos os dias. Essa tarefa é difícil, pois assim como a linguagem é ambígua, os sons produzidos também são e, às vezes, ruidosos.

Muitos são os algoritmos presentes na literatura, a exemplo dos apresentados por Jurafsky e Martin (2008) e Russell e Norvig (2009), cada qual adequado a um contexto. Portanto, cabendo o estudo de qual abordagem é mais adequada para cada aplicação.

3 TRABALHOS RELACIONADOS

Neste capítulo, é apresentado o mapeamento sistemático de literatura realizado para investigar as aplicações de ferramentas similares a que está sendo proposta e os recursos empregados nelas para resolver o problema. O protocolo de pesquisa do mapeamento sistemático realizado pode ser visto na seção 3.1. Os resultados obtidos a partir desse mapeamento podem ser vistos na seção 3.2. Na seção 3.3 são apresentadas as considerações do capítulo.

3.1 Protocolo de pesquisa

O protocolo de pesquisa desse mapeamento sistemático se baseia livremente no guia proposto por Petersen et al. (2008) e tem por objetivo investigar as aplicações práticas das ferramentas similares a que está sendo proposta, bem como os recursos empregados nelas para resolver o problema, como, por exemplo, algoritmos extratores de descritores e/ou outras informações semânticas de imagens e estruturas para o armazenamento dessas informações. Considerando esses objetivos, as seguintes questões foram definidas:

- Quais os métodos mais utilizados/eficientes para a extração dos descritores e metadados das imagens?
- Quais os métodos mais utilizados para a classificação dos descritores?
- Quais são as informações comumente agregadas aos descritores como metadados?

O processo de busca foi iniciado após a definição das bases de dados: A *Scopus*, por indexar muitas bases de dados e as bases *IEEE (Institute of Electrical and Electronics Engineers) Xplore Digital Library* e *ACM DL (Association Computing Machinery Digital Library)* por serem as bases de dados mais consolidadas no contexto da computação. A seguinte *string* de busca foi definida para mapeamento:

— "image"AND "based"AND "indoor"AND "objects"AND ("retrieval"OR "location")

Para a filtragem dos artigos relevantes e eliminação de falsos positivos foram definidos os seguintes critérios de seleção:

- **Inclusão:**
 - Trabalhos que propõem o uso de imagens para localizar objetos perdidos ou guardados ou catalogados;
 - Trabalhos que apresentem estratégias de extração de descritores e metadados de imagens de forma eficiente.

- **Exclusão:**

- Trabalho que não esteja disponível para *download*, de forma gratuita, através de bases de dados acessíveis pela Universidade Federal do Pampa (UNIPAMPA);
- Trabalho que não está escrito em português ou inglês;
- Trabalho com menos de quatro páginas;
- Anais completos de eventos.

Cada trabalho teve seus títulos, palavras-chave e resumos analisados para classificá-los. Os trabalhos escolhidos foram lidos, buscando obter um maior entendimento sobre os mesmos. Por fim, os trabalhos mais relevantes foram selecionados para compor o mapeamento sistemático.

3.2 Resultados da pesquisa

Inicialmente, com as condições citadas na seção anterior, foram retornados 10 trabalhos. Dentre estes, foram lidos e selecionados 9 trabalhos para compor o mapeamento sistemático, pois o décimo trabalho não estava disponível para acesso gratuito dentro da Unipampa.

No trabalho de He et al. (2015) é proposto um sistema de localização em ambientes internos na arquitetura cliente-servidor baseada em imagens previamente armazenadas e com informações de posição pré-calculadas por *Structure from Motion (SfM)*. Para a detecção do posicionamento é utilizado um processo iterativo de otimização do erro de posicionamento usando as imagens restantes. A fim de reduzir o número de operações a serem feitas é utilizada uma técnica de seleção nas imagens baseada na rejeição de quadros borrados ou desfocados. Para a extração dos descritores locais foi utilizado o algoritmo BFLoG, para os globais o RCFC, bem como o algoritmo PMVS para a geração de uma *Point Cloud* onde serão processados os *matchings*. Os resultados obtidos pelos autores apontam que o método implementado tem localização satisfatória, com 78% de precisão no algoritmo proposto para localização de, pelo menos, 5 metros, e com baixo tempo de resposta, podendo processar até 2 quadros por segundo.

No trabalho de Serrano, Savakis e Luo (2002) é proposto um sistema de classificação de imagens em ambientes internos e externos usando duas etapas: a primeira delas consiste na extração de características relacionadas às cores, as quais são processadas por uma máquina de vetor de suporte, e a extração de características relacionadas às texturas, decompostas para aumento de eficiência computacional, que é processada por uma outra. A segunda etapa se resume à aplicação dos resultados das duas máquinas de vetor de suporte anteriores a uma nova, que fará a efetiva classificação da imagem. Os resultados obtidos pelos autores apontam que mesmo com baixa complexidade e dimensionalidade é possível se atingir uma precisão satisfatória de 90.2%, podendo ser melhorada com outros elementos semânticos.

No trabalho de Suganthi S. Hanumanthappa (2016) é proposto um *framework* para a classificação de fotos de eventos acadêmicos ou de negócios. Para classificar as imagens são definidas 3 categorias em 2 níveis de computação: No primeiro nível, as imagens são classificadas em ambientes externos ou internos; No segundo nível, as imagens de ambientes externos são desconsideradas e as imagens de ambientes internos são classificadas em “com apresentação” ou “sem apresentação”. Para isso, é feita a extração de *features* através do histograma de cor no espaço HSV da imagem onde são computados média de contraste, média de uniformidade, entropia, dentre outras características a serem classificadas pelo algoritmo KNN. Os resultados obtidos indicam que as classificações são satisfatórias, tendo aproximadamente 94% de precisão em um universo de 635 imagens.

No Trabalho de Sivan e Darsan (2016) é proposta uma ferramenta assistiva para ajudar pessoas cegas a entenderem o ambiente ao redor e se localizar a partir de *feedback* auditivo. Para isso são utilizadas imagens de uma câmera e um *Raspberry PI* como plataforma para processar as imagens capturadas com a aplicação de várias técnicas de VC. Dentre as técnicas e ferramentas de VC aplicadas estão: Detecção de bordas usando *Canny Edge Detector*, extração de descritores usando BRISK, extração de texto das imagens utilizando *Tesseract OCR* e detecção de pessoas na cena usando *Haar Feature-based Cascade Classifier*. São empregados, como ferramentas computacionais, a biblioteca OpenCV e a linguagem de programação C++. Os resultados obtidos pelos autores são considerados bons e podem ser aprimorados com dispositivos melhores e mais compactos no futuro, não sendo necessário descartar quadros importantes do vídeo por estarem borrados ou desfocados.

No trabalho de Mighali et al. (2015) é proposto um sistema de informação para museus inteligentes, de conteúdo personalizado e de experiências compartilháveis (redes sociais). Para isso, o perfil de cada visitante é obtido a partir de informações capturadas por um dispositivo vestível que é disponibilizado pela equipe do museu e processadas em tempo real. O vestível conta com uma câmera e um sensor *bluetooth* de baixo consumo energético que é capaz de calcular a posição atual a partir da distância dos pontos de referência usando a intensidade do sinal. A câmera, por sua vez, faz a captura das imagens que o usuário está vendo, indicando ao sistema que obra do museu está sendo observada, possibilitando assim a personalização do conteúdo apresentado através do smartphone, onde se pode compartilhar o conteúdo, ou através de tecnologias de Internet das coisas, do inglês *Internet of Things* (IOT) espalhadas pelo museu. Para processar as informações das imagens é usado o extrator de descritores ORB e é usado o algoritmo de RANSAC para a rejeição de *outliers*. Os resultados obtidos indicam a viabilidade da aplicação dessas tecnologias em conjunto nesse contexto, sendo que as validações em um ambiente real poderão ser feitos em trabalhos futuros.

No trabalho de Guo et al. (2018) é proposto um sistema de localização em ambientes internos usando processamento de imagens de câmeras estéreo e *WiFi fingerprinting*

para geração de modelos tridimensionais de ambientes a partir de pontos e localizações definidas. A partir dessa base e de uma foto tirada pelo usuário com uma câmera, em geral de *smartphone*, é estimada a posição do mesmo. A extração dos descritores das imagens é feita usando o ORB e aplicando a abordagem Multi-Probe LSH no algoritmo KNN de classificação para a correspondência, bem como *Fundamental Matrix* para reduzir o número de resultados equivocados. Os resultados obtidos indicam que o sistema localiza com sucesso de 96% em uma cena e 98% na outra, tendo um erro de direção de 6° e de localização de menos de 20cm em 30% e de menos de 50cm em 60% dos casos.

No trabalho de Kabbai, Abdellaoui e Douik (2019) é proposto um novo extrator de características baseado na concatenação de *features* locais e globais chamado CLCG. Para isso foi necessário o uso de uma série de adaptações em técnicas e estratégias no estado da arte, ao exemplo da implementação de um variante do *Local Ternary Pattern* (LTP) de menor dimensionalidade, nomeadamente *Elliptical upper and lower local ternary pattern* (EUL LTP) para os descritores globais. Para os descritores locais foi utilizado o algoritmo SURF com as *features* divididas em *codebooks* por um *k-means*. O resultado apresentado pelos autores é promissor, sendo ligeiramente melhor que muitos dos métodos no estado da arte e consideravelmente próximo de métodos envolvendo *deep neural networks*.

No trabalho de Liu e Zhang (2012) é proposto um sistema de localização de ciclos em trajetórias de robôs usando *matching* de *features* extraídas usando o algoritmo SIFT e organizando-as em uma árvore K-dimensional. Para não haver uma carga desnecessária de processamento de *frames* iguais é feita uma seleção de *keyframes* a serem processados. Os resultados obtidos indicam que o método proposto é consideravelmente mais rápido que BoW mas ainda não aplicável em grande escala devido ao nível de complexidade $\mathcal{O}(n \log n)$

O trabalho de Petraitis et al. (2017) descreve uma implementação de classificador de imagens usando BoW de “*Visual Words*” usando máquinas de vetor de suporte como classificadores para vários tipos de cenas. Para tanto, as classificações são realizadas em níveis superiores de dimensionalidade e, em seguida, feita uma nova classificação caso a primeira não tenha sido conclusiva. Além disso, o trabalho ainda traz uma análise estatística de quais são os algoritmos mais precisos e mais rápidos no contexto analisado. Os resultados obtidos indicam que dentre os algoritmos avaliados o algoritmo SURF foi o mais preciso nos testes, com 85% e o USURF perdendo em 2 pontos percentuais, porém com um ganho de 33% em tempo de execução.

Após leitura aprofundada dos trabalhos selecionados, as questões levantadas foram respondidas de forma sintetizada na Tabela 1.

3.3 Considerações do capítulo

Através do mapeamento sistemático realizado, pode-se perceber um número crescente de estratégias que podem ser aplicadas na classificação e detecção de imagens. Nesse

Tabela 1 – Trabalhos Relacionados

Título do Trabalho	Autores	Q1	Q2	Q3
Efficient Image Retrieval Based Mobile Indoor Localization	(HE et al., 2015)	BFLoG e RCFC	PMVS (Point Cloud)	Posição Relativa da Câmera
A Computationally Efficient Approach to Indoor/Outdoor Scene Classification	(SERRANO; SAVAKIS; LUO, 2002)	Histograma e Wavelet Transform	SVM	Não Se Aplica
Classification of Event Image Set Using Mining Techniques	(SUGANTHI S. HANUMANTHAPPA, 2016)	Histograma (HSV)	KNN	-
Computer Vision based Assistive Technology For Blind and Visually Impaired People	(SIVAN; DARSAN, 2016)	BRISK	BRISK	Texto Contido na Imagem
Innovative IoT-aware Services for a Smart Museum	(MIGHALI et al., 2015)	ORB	Não Descrito	Posição Relativa a Intensidade do Sinal
Fusion of WiFi and Vision based on Smart Devices for Indoor Localization	(GUO et al., 2018)	ORB	KNN	Posição Relativa a Intensidade do Sinal
Image classification by combining local and global features	(KABBAI; ABDELLAOUI; DOUIK, 2019)	CLCG	K-Means (BoW)	-
Indexing Visual Features: Real-Time Loop Closure Detection Using a Tree Structure	(LIU; ZHANG, 2012)	SIFT	K-Means (KD-Tree)	-
Environment Recognition based on Images using Bag-of-Words	(PETRAITIS et al., 2017)	Varios	SVM	Categoria

Fonte: Próprio Autor

contexto, trabalhos como o de Tareen e Saleem (2018), o de Petraitis et al. (2017), e o de Lyra (2018), dentre outros, são importantes para ter uma visão de em quais contextos cada estratégia é válida.

Nos artigos avaliados, com maior correlação com o trabalho proposto, nota-se uma tendência forte no uso de redes sem fio para estimar localizações por meio de cálculos usando intensidade de sinal. No entanto, para isso, é necessário que haja uma quantidade de dispositivos de referência fora dos padrões de um cidadão comum (que é de um único

roteador por residência).

Durante o processo de busca, nenhum trabalho com o intuito de localizar objetos perdidos, guardados ou catalogados usando imagem, texto e voz foi encontrado. No entanto, o trabalho de Sivan e Darsan (2016) apresenta a maior proximidade com nossa proposta, utilizando VC e reconhecimento de texto para definir os objetos de interesse, tendo sido uma referência inicial para a obtenção dos objetivos deste trabalho.

4 MATERIAIS E MÉTODOS

Neste capítulo, serão apresentados todos os processos de desenvolvimento, arquitetura e infraestrutura da aplicação proposta. Na seção 4.1 serão exibidos o diagrama de pacotes e de fluxo da aplicação bem como as respectivas responsabilidades de cada módulo da aplicação. Na seção 4.2 e seção 4.3 serão apresentadas as tecnologias aplicadas para cada um dos diferentes módulos da aplicação bem como se dá a comunicação entre eles. Na seção 4.4 será abordado o ambiente em que a aplicação foi desenvolvida e testada. Na seção 4.5 será tratado o processo de desenvolvimento adotado durante os trabalhos. Na seção 4.6, por fim, será apresentado o protótipo desenvolvido.

4.1 Arquitetura da aplicação

A arquitetura geral da aplicação está definida em duas partes distintas: A primeira delas sendo a parte referente ao cliente, uma aplicação móvel no SO Android; A segunda delas sendo a parte referente ao servidor, uma aplicação em Python para o SO Linux. O fluxo de funcionamento da aplicação é apresentado na Figura 6, tratando separadamente o cliente e o servidor.

4.1.1 Aspectos do cliente

- Captura do conteúdo audiovisual;
- Captura da informação textual;
- Captura das informações de redes *wifi* e geográficas;
- Sistema de comunicação com o servidor.

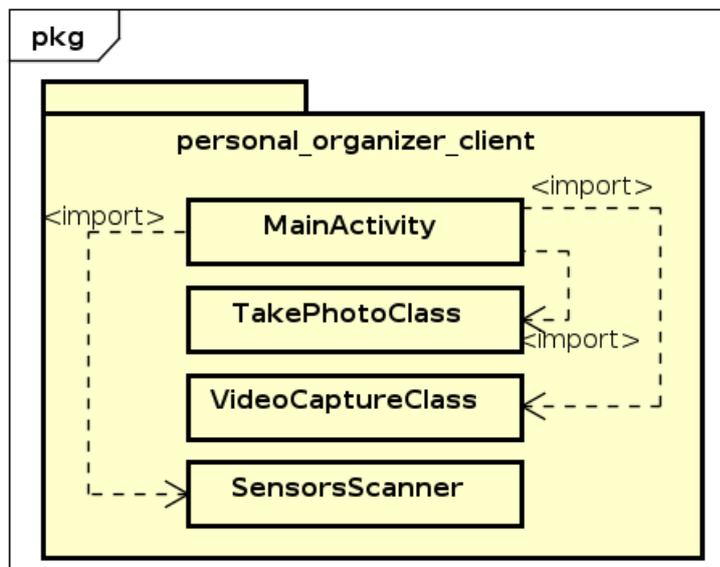
O diagrama de pacotes que representa esses aspectos está apresentado na Figura 4.

4.1.2 Aspectos do servidor

- Servidor http com acesso a um banco de dados;
- Reconhecimento das palavras faladas do áudio;
- Rastreador do objeto no vídeo;
- Árvore de vocabulário.

O diagrama de pacotes que representa esses aspectos está apresentado na Figura 5.

Figura 4 – Diagrama de pacotes do cliente.



Fonte: Próprio Autor

4.2 Infraestrutura da aplicação com relação ao servidor

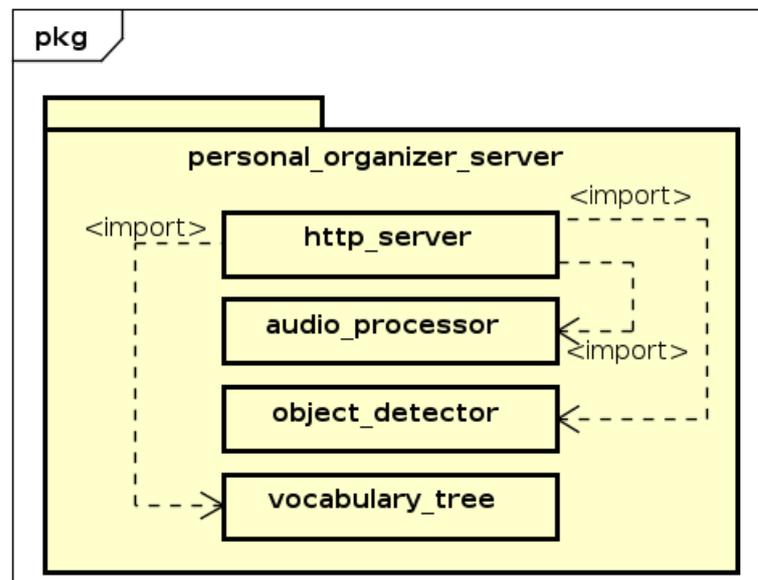
A tecnologia adotada para a construção da comunicação *http* foi um *framework* minimalista de desenvolvimento web da linguagem Python, conhecido como Flask, que foi utilizado na classe *http_server* para realizar o controle do servidor, bem como a comunicação com o banco de dados.

A conversão de voz em texto é realizada por um serviço de inteligência artificial e aprendizagem de máquina chamado Google *Text-to-Speech*, que é responsável, por meio de chamadas de API, por converter o áudio extraído do vídeo em texto legível com os respectivos *timestamps* de cada palavra¹. A classe responsável por esta operação é a *audio_processor*.

A sinalização de entrada do objeto na cena é de responsabilidade do usuário, a partir de comando de voz reconhecido pelo Google *Text-to-Speech*. Em seguida, o rastreamento é iniciado por meio de um método da biblioteca *dlib*, implementado com base na proposição de Danelljan et al. (2014). Em adicional, para a detecção da saída do objeto de cena, é necessário a extração de características por meio do método ORB, implementado pela biblioteca OpenCV, pois, os métodos de rastreamento testados durante o desenvolvimento, não se mostraram sensíveis ao desaparecimento de objetos que não cruzam as margens da cena. A classe responsável pela realização do rastreamento e detecção da saída do objeto é a *object_detector*.

¹ Apesar da ferramenta se chamar Google *Text-to-Speech*, além de conversão de texto em voz, ela também realiza conversão de voz em texto

Figura 5 – Diagrama de pacotes do servidor.



Fonte: Próprio Autor

A árvore de vocabulário, estrutura responsável por armazenar de forma hierárquica as características extraídas dos objetos, foi implementada de forma amplamente inspirada no trabalho de Nister e Stewenius (2006), com partes do código adquiridas diretamente do trabalho de conclusão de Lyra (2018), publicada de forma livre e indiscriminada. A estrutura de árvore de vocabulário utiliza, além da linguagem de programação Python, o C++, dado que, de acordo com o autor, o Python é uma linguagem com limitações quanto à gerência de memória. A classe responsável por controlar a estrutura de árvore definida é a *vocabulary_tree*.

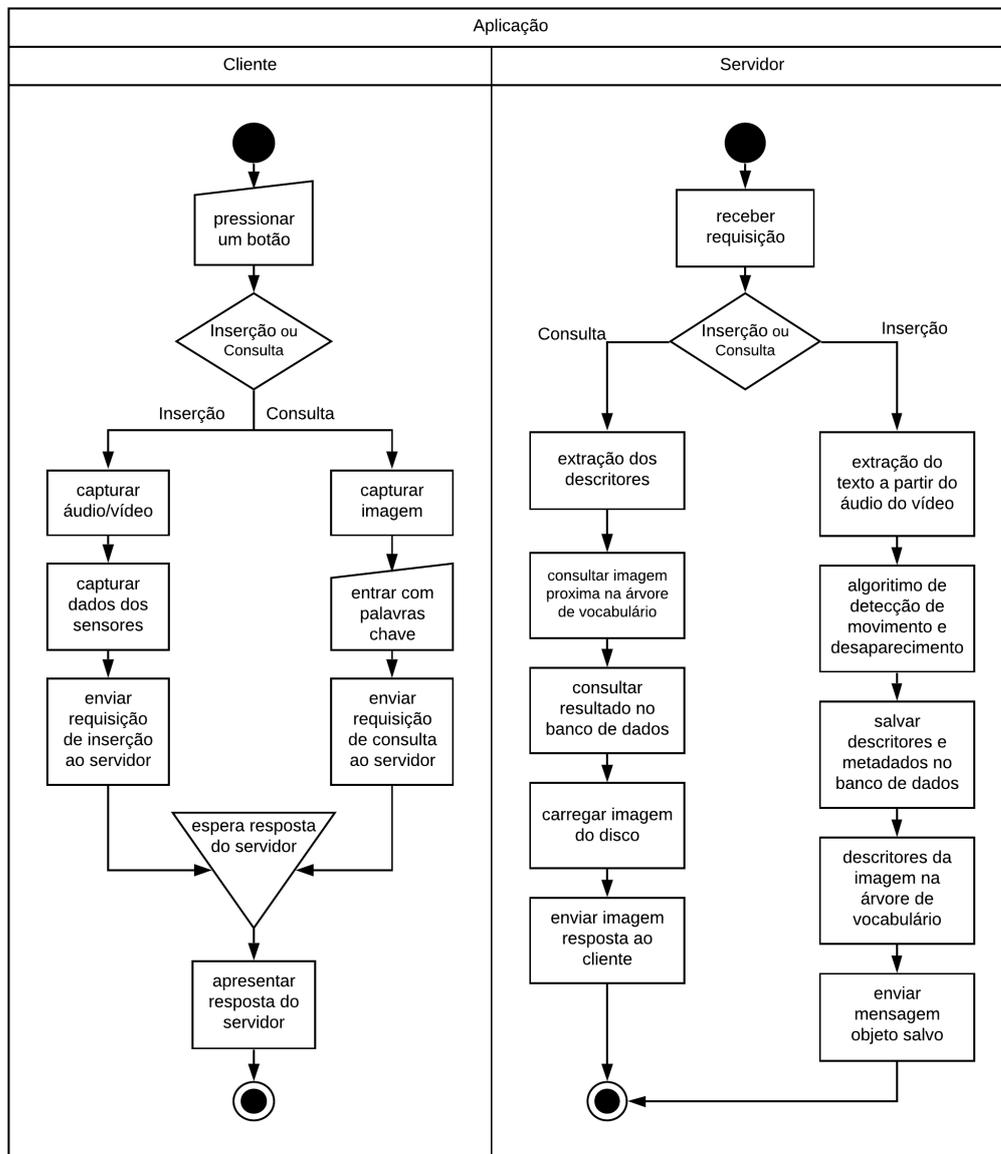
4.3 Infraestrutura da aplicação com relação ao cliente

Para a construção do cliente da aplicação foram necessárias a utilização de bibliotecas próprias já presentes no ambiente de desenvolvimento do Java Android, como o *WirelessManager* responsável por extrair as informações de redes *wifi* disponíveis e o *SensorsManager* responsável pela leitura de sinais de sensores necessários para o cálculo do posicionamento.

Em adicional foi necessário utilizar a biblioteca externa OkHTTP para que fosse simplificada a comunicação via *http*, recurso de vital importância para a comunicação cliente servidor o qual não existe de maneira facilmente aplicável nativamente na respectiva plataforma.

O uso do cliente da aplicação é realizado, quanto a inserção, a partir do início

Figura 6 – Diagrama de fluxo da aplicação:



Fonte: Próprio Autor

da captura das imagens a partir de um botão, quando o objeto já está devidamente centralizado na câmera e o foco devidamente ajustado, a partir de então a captura das imagens é realizada. Quando a interrupção da captura é feita, também através de um botão, as imagens captadas são enviadas ao servidor no formato de um vídeo, juntamente com as informações das leituras dos sensores.

O uso do cliente da aplicação é realizado, quanto à consulta, a partir da captura de uma imagem a partir de um botão, quando o objeto já está devidamente centralizado na câmera e o foco devidamente ajustado, a partir de então é apresentado uma caixa de

diálogo ao usuário na qual é disponibilizado a oportunidade de inserir palavras chave para a busca de seus objetos, caso queira.

As ações de envio e recebimento de mensagens HTTP, captura do texto digitado e detecção do pressionamento dos botões é de responsabilidade do módulo *MainActivity*, este por sua vez faz a captura do vídeo ou imagem, através dos módulos *TakePhotoClass* ou *VideoCaptureClass*, e dos sinais dos sensores, através do módulo *SensorsScanner*.

4.4 Ambiente de execução da aplicação

O dispositivo responsável pela execução do servidor de teste da aplicação é um *desktop* com a seguinte configuração:

- Sistema Operacional Linux - Pop OS 20.04;
- Python 3.8;
- Placa Mãe Asrock A320M-HD;
- Processador AMD Ryzen 5 1600 AF;
- Placa de vídeo Radeon RX 570;
- 16 Gigabytes de memória RAM;
- 128 Gigabytes de SSD Sandisk SATA 3.

O dispositivo responsável pela execução do cliente de teste da aplicação é um *smartphone* Xiaomi com a seguinte configuração:

- Sistema Operacional Linux - Android 10;
- Chipset Snapdragon 660;
- Placa de vídeo Adreno 512;
- 4 Gigabytes de memória RAM;
- 64 Gigabytes de armazenamento.

4.5 Processo de desenvolvimento

Devido à natureza de alta complexidade do processo de desenvolvimento da aplicação, bem como o uso de uma ampla gama de tecnologias, foram utilizados princípios de Métodos Ágeis e do Scrum, envolvendo reuniões semanais de avaliação das ferramentas e do desenvolvimento com o professor orientador. Também foi realizada a mudança dos

requisitos conforme fosse necessário para o não interrompimento do processo de desenvolvimento do presente trabalho, por vezes sendo utilizada a técnica de *Brainstorming* para a decisão dos passos seguintes. As ferramentas utilizadas para o desenvolvimento estão listadas a seguir.

- Sublime Text como editor de texto para o código fonte do servidor;
- Sublime Merge como ferramenta para controle de versão;
- Github como plataforma de armazenamento do código fonte;
- Terminal Bash para a execução do servidor da aplicação;
- Android Studio como IDE para a construção do aplicativo cliente.

4.6 Protótipo desenvolvido

De acordo com a definição da arquitetura e das tecnologias, o protótipo foi construído de forma a ser o mais simples possível de ser utilizado, bastando a inicialização do servidor em uma máquina com sistema operacional Linux, com a linguagem Python instalada, fazer o *download* das dependências a partir do arquivo de requerimentos, carregar as credenciais do Google Cloud, inicializar o servidor e instalar o aplicativo no dispositivo Android. Devido ao protótipo ter sido pensado para ser expansível, o sistema foi elaborado de forma que cada módulo tenha uma responsabilidade, como pode ser visto na Figura 4.

A Figura 7 apresenta a saída padrão do servidor em funcionamento.

O funcionamento do algoritmo proposto para a detecção de movimentação e desaparecimento do objeto da cena se dá através de uma função iterativa que, inicialmente, extrai as características do objeto que foi centralizado na imagem baseado na regra dos terços², como pode ser visto na Figura 9, e então, avalia quadro a quadro do vídeo capturado, redimensionando a imagem, se necessário, realizando a normalização do histograma da iluminação a partir do algoritmo CLAHE e utilizando o extrator de características ORB, onde, caso haja mais de 20% das características capturadas inicialmente, aplica-se o rastreamento de correlação (DANELLIAN et al., 2014) para saber a posição atual do objeto na próxima iteração, caso contrário, captura o quadro de saída do objeto da cena. A Figura 8 apresenta uma ilustração do fluxo do algoritmo desenvolvido.

O desenvolvimento do sistema de localização em ambientes interno mostrou-se inviável de ser realizado no tempo designado, conjuntamente com as demais funcionalidades. No protótipo desenvolvido neste trabalho, no entanto, os identificadores e as intensidades

² Regra dos Terços é uma técnica utilizada na fotografia para se obter melhores resultados. Para utilizá-la deve-se dividir a fotografia em 9 quadrados, traçando 2 linhas horizontais e duas verticais imaginárias, e posicionando nos pontos de cruzamento o assunto que se deseja destacar para se obter uma foto equilibrada.

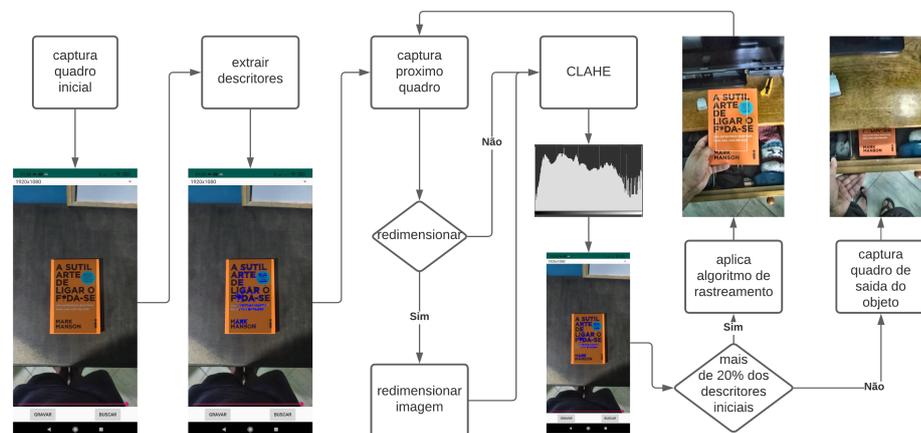
Figura 7 – Captura da tela do servidor

```
wesley@pop-os: ~/TCC/personal-organizer-server
wesley@pop-os:~/TCC/personal-organizer-server$ export GOOGLE_APPLICATION_CREDENTIALS="./credentials.json"
wesley@pop-os:~/TCC/personal-organizer-server$ sudo docker-compose up -d db
[sudo] senha para wesley:
personal-organizer-server_db_1 is up-to-date
wesley@pop-os:~/TCC/personal-organizer-server$ source env/bin/activate
(env) wesley@pop-os:~/TCC/personal-organizer-server$ python http_server.py
(env) wesley@pop-os:~/TCC/personal-organizer-server$ python http_server.py
tabela de imagens não existe, criando
* Serving Flask app "http_server" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
* Restarting with stat
tabela de imagens existe, importando
* Debugger is active!
* Debugger PIN: 960-180-681
MoviePy - Writing audio in files/WF20201128222705/WF20201128222705.wav
MoviePy - Done.

{'capturar': {'start': datetime.timedelta(0), 'end': datetime.timedelta(seconds=1, microseconds=600000)}, 'livro': {'start': d
atetime.timedelta(seconds=1, microseconds=600000), 'end': datetime.timedelta(seconds=1, microseconds=900000)}, '7': {'start':
datetime.timedelta(seconds=1, microseconds=900000), 'end': datetime.timedelta(seconds=2, microseconds=400000)}}
buildando a arvore
20201128222705, 15285346533
127.0.0.1 - - [28/Nov/2020 22:27:20] "POST / HTTP/1.1" 200 -
'capturar': WF20201128222705, 8784007
127.0.0.1 - - [28/Nov/2020 22:27:20] "POST /search HTTP/1.1" 200 -
(0.10379761457443237, '20201128222705')
21168110
127.0.0.1 - - [28/Nov/2020 22:27:20] "POST /search HTTP/1.1" 200 -
MoviePy - Writing audio in files/WF20201128222720/WF20201128222720.wav
MoviePy - Done.
```

Fonte: Próprio Autor

Figura 8 – Ilustração do algoritmo de detecção de movimentação e desaparecimento.



Fonte: Próprio Autor

de cada um dos sinais de redes *wireless*, bem como o ângulo da bússola, são capturados e armazenados no banco de dados para tratamento posterior.

Figura 9 – Livro centralizado, a esquerda, e em detalhes, com os *keypoints* do ORB, na direita.



Fonte: Próprio Autor

5 RESULTADOS E DISCUSSÕES

No presente capítulo, serão discutidos os procedimentos de validação da aplicação na seção 5.1, os resultados obtidos a partir da validação na seção 5.2, análise e discussão dos resultados obtidos na seção 5.3.

5.1 Procedimento de validação

O procedimento de validação do protótipo foi separado em quatro etapas distintas: a primeira etapa refere-se à avaliação do procedimento de inserção dos objetos no protótipo desenvolvido de forma manual; a segunda etapa apresenta a avaliação do desempenho da detecção da saída de cena do objeto; na terceira etapa, apresenta a avaliação dos resultados da busca visual e a quarta etapa, apresenta a avaliação do tempo de inserção e consulta de forma automatizada. Para os testes, foram selecionados cinco objetos e quatro lugares, totalizando vinte combinações distintas.

Para a validação da primeira etapa, foi executada a inserção no sistema de dez vezes cada uma das vinte combinações, para que se tivesse segurança em relação aos resultados apresentados para cada uma delas. Cada uma das inserções é salvo com um código identificador único, para que fosse possível a análise posterior, identificando sucessos e fracassos.

Para a avaliação da segunda etapa, foi executada uma busca, a partir das palavras faladas na inserção da etapa anterior, identificando as imagens relativas aos códigos únicos e categorizando-as em cinco casos de detecção diferentes, dado que uma classificação binária não representaria de forma satisfatória, por tratar-se de uma análise qualitativa de imagens.

Os casos de categorização são:

- **Resultado Inconclusivo:** Quando não houver imagem de evento de saída de cena ou não for possível identificar nem o objeto nem o local na imagem;
- **Resultado Insatisfatório:** Quando na imagem do evento de saída do objeto de cena for possível identificar de forma parcial o objeto ou o local, por ter detectado antecipada ou tardiamente o desaparecimento;
- **Resultado Aceitável:** Quando na imagem do evento de saída do objeto de cena for possível identificar visualmente de forma parcial o objeto ou o local por estar fora de foco;
- **Resultado Bom:** Quando na imagem do evento de saída do objeto de cena for possível identificar visualmente de forma clara o objeto ou o local;
- **Resultado Excelente:** Quando na imagem do evento de saída do objeto de cena for possível identificar visualmente de forma clara o objeto e o local.

A Figura 10, Figura 11 e Figura 12 são alguns dos exemplos das inserções dos testes realizados, sendo categorizados respectivamente como um excelente, um aceitável e um insatisfatório.

O caso de teste da Figura 10 foi categorizado como excelente, pois, percebe-se nitidamente o objeto, um livro, e o local onde ele foi guardado, uma gaveta.

O caso de teste da Figura 11 foi categorizado como aceitável, apesar do objeto, livro, ser visível na cena, o desfoque torna difícil extrair informação útil, assim como não é possível saber, com certeza, onde o objeto foi guardado.

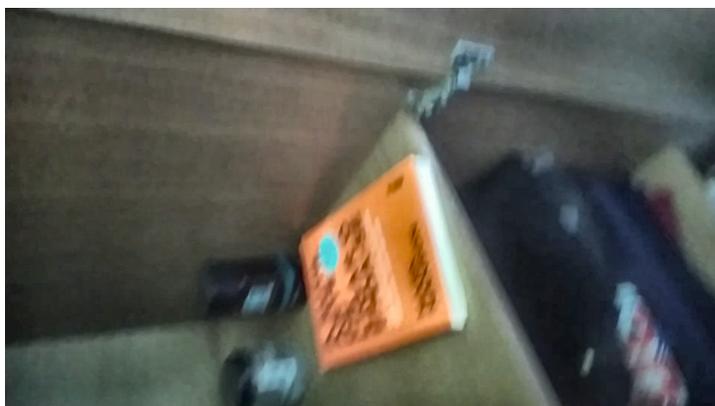
O caso de teste da Figura 12 foi categorizado como insatisfatório, pois, nota-se de forma bem abstrata, que um objeto foi armazenado em uma prateleira, no entanto, o próprio objeto é praticamente imperceptível e a saída de cena foi detectada muito tarde.

Figura 10 – Caso de teste considerado excelente.



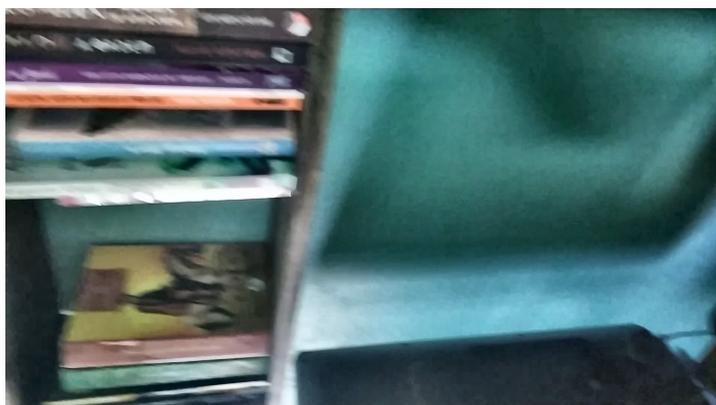
Fonte: Próprio Autor

Figura 11 – Caso de teste considerado aceitável.



Fonte: Próprio Autor

Figura 12 – Caso de teste considerado insatisfatório.



Fonte: Próprio Autor

Para a realização da terceira etapa foi executada uma busca, a partir de imagens exemplo de cada um dos cinco objetos testados no banco de dados construído na primeira etapa. As imagens de exemplo podem ser vistas na Figura 13.

Figura 13 – Objetos utilizados nos testes.



Fonte: Próprio Autor

Para a realização da quarta etapa, foi executado um teste automatizado onde foram registrados trezentas vezes, de forma aleatória, as inserções anteriormente feitas, para a avaliação dos tempos de execução conforme o aumento do banco de dados do sistema.

5.2 Resultados obtidos

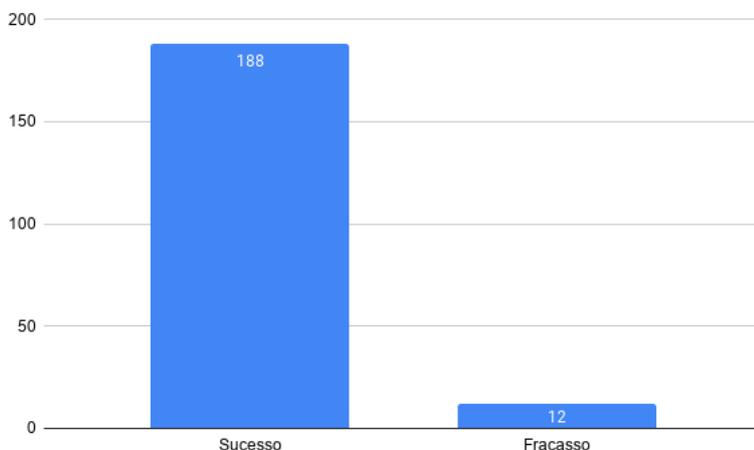
5.2.1 Resultados da etapa 1

A partir dos dados obtidos na primeira etapa dos testes, que podem ser observados no Apêndice A, foi montado o gráfico da quantidade de sucessos e fracassos de inserção da Figura 14, e também o gráfico com os tempos de cada inserção da Figura 15, dos duzentos testes.

Os dados apresentados na Figura 14 indicam que a cada 200 tentativa de inserção no sistema, 94%(188) é bem sucedidas em todas as etapas do processo, enquanto que, 6%(12) delas falham por problemas de comunicação entre o cliente e o servidor, na detecção do rastreamento ou do desaparecimento.

Os dados apresentados na Figura 15 apontam que, em média, o tempo das detecções e inserções é de 18,5 segundos, o desvio padrão, por sua vez, fica em torno de 5,2 segundos.

Figura 14 – Quantidade de sucessos e fracassos para os duzentos testes realizados.

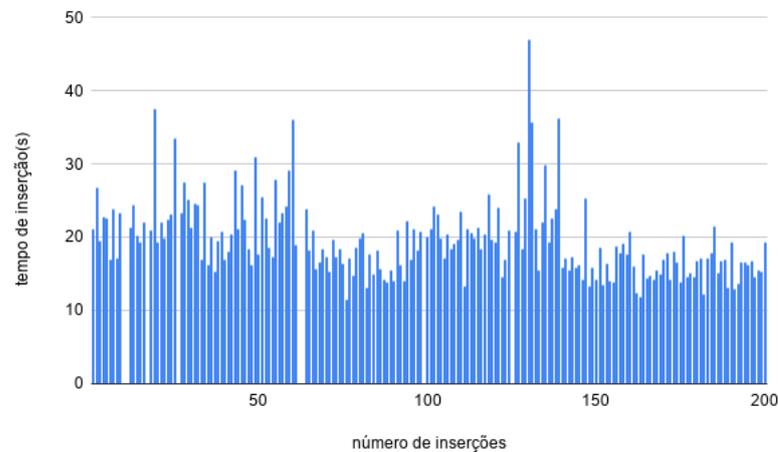


Fonte: Próprio Autor

5.2.2 Resultados da etapa 2

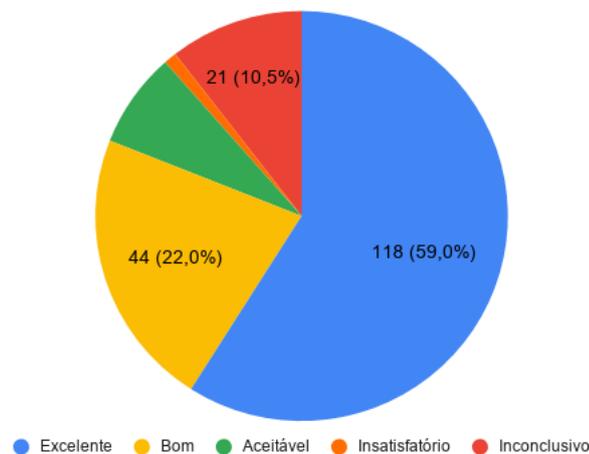
A partir dos dados obtidos na segunda etapa, que podem ser vistos no Apêndice B, foi montado um gráfico com os casos de categorização definidos nos procedimentos, apresentado na Figura 16, bem como o gráfico com os tempos de consulta textual, apresentado na Figura 17.

Figura 15 – Tempo em segundos de cada uma das inserções.



Fonte: Próprio Autor

Figura 16 – Porcentagem de cada um dos casos de categorização.



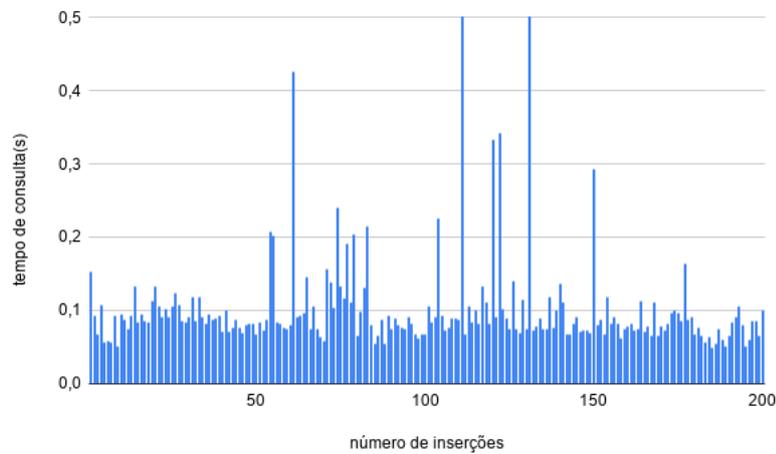
Fonte: Próprio Autor

5.2.3 Resultados da etapa 3

A partir dos dados obtidos na terceira etapa do teste, como pode se perceber na Tabela 2, em uma mesma configuração de árvore, os diversos objetos têm tempos de consulta diferentes. Acredita-se, analisando as imagens, que isso tenha a ver com o foco da câmera e a capacidade do algoritmo ORB de descritores em certas circunstâncias de luz e sombra. A média e variância dos tempos das consultas de cada objeto podem ser vistos na Figura 18.

Pode-se notar também, nesta etapa do teste, que a árvore de vocabulário utilizada,

Figura 17 – Tempos em segundos da consulta textual.



Fonte: Próprio Autor

Tabela 2 – Tabela dos tempos em nanosegundos das dez consultas visuais de cada objeto

	Livro	Kindle	Remédio	Baralho	Controle
1	420753743	248547292	316874215	108481374	136711645
2	297509698	222339281	305173823	92312293	136595443
3	298150549	228014339	302688067	97613045	144875344
4	310676362	209907223	303014259	97009586	137860980
5	304889745	202986093	317864350	94782676	132336916
6	307095663	223797643	304111369	94916587	137880569
7	284411257	206015811	305048660	110049333	142706254
8	293404771	213065399	305476763	96480208	124267376
9	289568735	207301046	303961132	93074020	131519782
10	300047538	285684237	305735634	91359807	141773700

Fonte: Próprio Autor

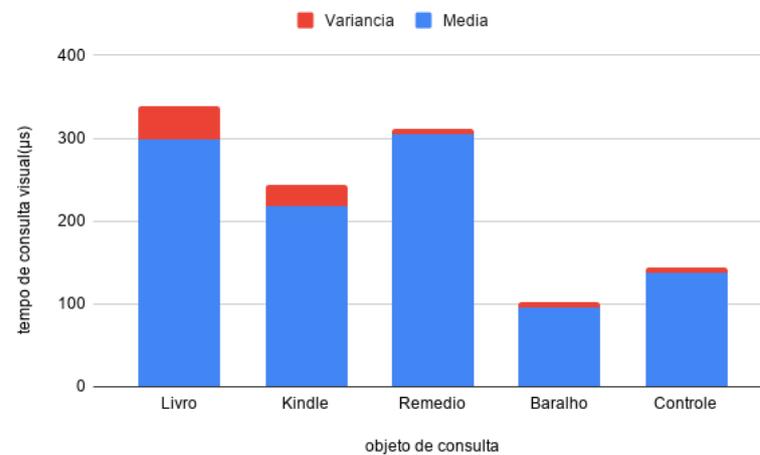
algoritmo utilizado para o armazenamento e consulta das imagens, tem suas respostas atemporais, ou seja, não leva em conta quando o objeto foi inserido, assim como não leva em conta quaisquer outros dados que não sejam descritores de imagens.

5.2.4 Resultados da etapa 4

A partir dos dados obtidos na quarta etapa do teste, dividida em duas séries de testes, uma com e uma sem o reagrupamento dos descritores da árvore de vocabulário, pode-se perceber dois aspectos relevantes para o entendimento das limitações da tecnologia escolhida para este protótipo.

A primeira delas, como pode ser visto na Figura 19 e na Figura 20, é que tanto o tempo de inserção quanto a variância deste crescem rapidamente, exigindo reagrupamento

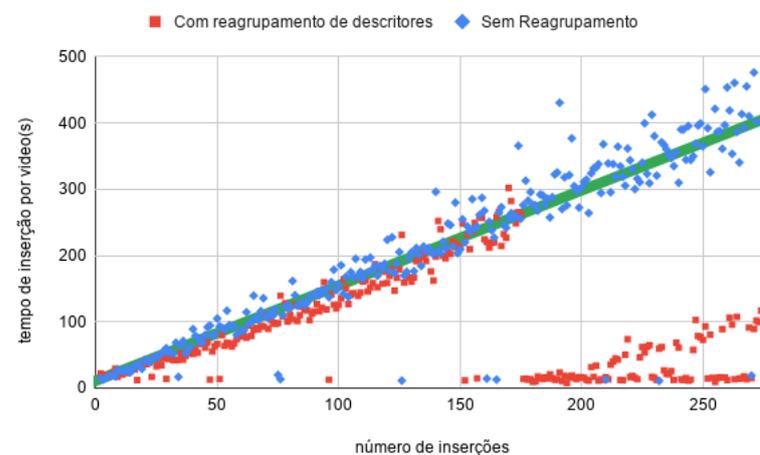
Figura 18 – Tempo da consulta visual do teste manual.



Fonte: Próprio Autor

dos descritores na árvore de vocabulário constantemente, para que a solução mantenha um bom nível de desempenho. O impacto do reagrupamento dos descritores pode ser visto nestas figuras, e, também, no gráfico das inserções manuais apresentadas na Figura 15, onde os reagrupamentos eram realizados a cada dez testes, que apontam uma média de 18,5 segundos de tempo de inserção.

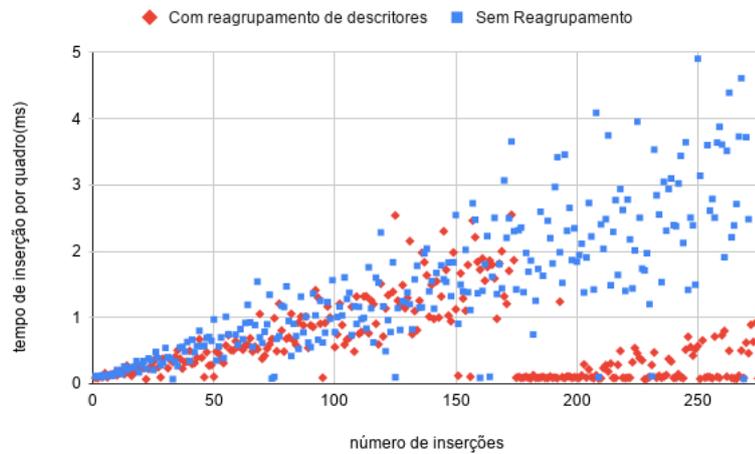
Figura 19 – Tempo em segundos de inserção por vídeo.



Fonte: Próprio Autor

A segunda diz respeito aos tempos de consulta, que assim como a inserção têm um aumento da variância com o crescimento do conjunto de testes. No entanto, diferentemente da inserção, os tempos de consulta não sofrem grandes impactos pelo reagrupa-

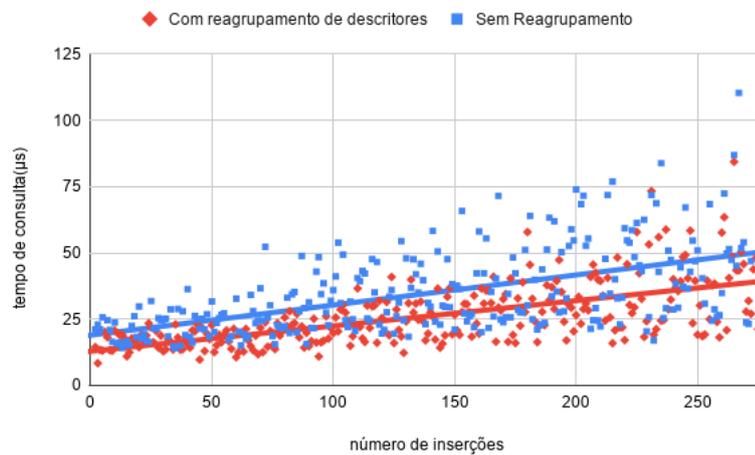
Figura 20 – Tempo em segundos de inserção por quadro.



Fonte: Próprio Autor

mento dos descritores da árvore, apenas uma diminuição leve da variância dos tempos. Esses dados podem ser vistos na Figura 21.

Figura 21 – Tempo da consulta visual do teste automatizado.



Fonte: Próprio Autor

5.3 Análise dos resultados

Em decorrência do que foi demonstrado na seção anterior e observando-se as imagens de saída dos objetos da cena, pode-se perceber que os resultados apresentados são promissores, visto que na Figura 16, combinando as classificações boas e excelen-

tes, foi identificado 81% de sucesso, contra os 12,5% observados nos insatisfatórios ou inconclusivos, mostrando-se, o restante, apenas como resultados aceitáveis.

Quanto ao tempo de processamento da aplicação, devido ao rápido crescimento do tempo de execução da inserção da árvore de vocabulário, não é viável de ser realizado em tempo real, se fazendo necessário o processamento assíncrono e necessitando o reagrupamento constante dos descritores para que o processo seja minimamente viável, funcionalidade que não foi possível de ser realizado em tempo hábil para o presente trabalho.

Em termos da consulta textual, como era de se esperar, dada a robustez dos bancos de dados relacionais, os resultados apresentam uma média de 80 milissegundos para a recuperação dos objetos, sendo suficientemente rápido para uma pesquisa.

A consulta visual, por sua vez, apresenta bons tempos de carregamento, no entanto, devido à limitação da árvore de vocabulário utilizada não suportar metadados, ainda não há como saber qual das inserções da lista retornada é a mais recente, podendo ser um tópico abordado em trabalhos futuros.

A implementação da árvore de vocabulário utilizada neste protótipo, além de não suportar metadados, tem também problemas de memória e tempo de execução das inserções, em especial se referindo ao uso de descritores binários, como os retornados pelo algoritmo ORB utilizado neste protótipo, portanto não foi possível realizar nenhum teste com mais de trezentas inserções.

6 CONCLUSÃO

Buscando estudar tecnologias e técnicas que gerem bons resultados e tornem viável a produção de uma ferramenta de organização e localização de acervos em ambientes internos, este trabalho teve como objetivo construir um protótipo de uma aplicação para este fim, por meio do emprego de tecnologias como VC, processamento de texto e voz e localização interna por meio de dispositivos sem fio. Acredita-se que, com as tecnologias estudadas, pode-se transformar o protótipo criado em um produto, beneficiando a todos.

A realização deste trabalho foi um grande desafio em vários aspectos: técnicos, de tempo e até mesmo emocionais. A princípio não se sabia exatamente quais tecnologias seriam empregadas e muitas perguntas estavam em aberto, ainda mais, após uma revisão de literatura que retornou resultados ligeiramente distintos do esperado, o que acabou contribuindo muito pouco para a escolha final das tecnologias aplicadas.

No decorrer do presente trabalho, alguns desafios, como, selecionar o que focar e o que deixar de lado, definir ferramentas, métodos e como integrá-los, falhas com o servidor da aplicação e de conversão de voz em texto, que foram sanados, interferiram no desenvolvimento do protótipo.

Apesar dos obstáculos, o protótipo alcançou um resultado satisfatório ao que se propõe, apresentando mais de 80% de sucesso, e tempos que, apesar de não serem ideais, aparentemente, não comprometem a usabilidade.

O trabalho realizado contribuiu significativamente para o crescimento pessoal e profissional do autor, agregando conhecimentos de diversas tecnologias que não foram amplamente abordadas durante o período de curso, sejam elas visão computacional, processamento de linguagem natural dentre outras.

O protótipo proposto, apesar de ter apresentado um resultado satisfatório, demonstra uma ampla gama de melhorias que podem ser aplicadas, tais como:

- Em termos da estrutura de dados, tornando a árvore de vocabulário auto-ajustável, não exigindo desligamento do servidor para isso e tornando-a salvável e adaptável a metadados;
- Em termos de otimização, reduzindo os tempos de processamento das informações a partir de pesquisa e desenvolvimento de novos métodos e resolvendo os problemas de memória da linguagem Python;
- Em termos de dependências externas para que a ferramenta possa ser inteiramente controlável e adaptável para o contexto aplicado.
- Em termos de métodos e técnicas, podendo substituir as configurações hoje realizadas de forma manual, como o ajuste de foco e o início da captura, por algoritmos inteligentes

Além disso, a ferramenta apresenta também, um problema prático com vários aspectos a serem cientificamente investigados nas áreas abordadas neste trabalho, bem como a possibilidade de geração de propriedade intelectual e desenvolvimento de resultados práticos tecnologicamente relevantes.

REFERÊNCIAS

- ALPAYDIN, E. **Introduction to machine learning**. 2. ed. Cambridge, MA, USA: MIT press, 2009. Citado na página 31.
- AMAZON.COM, INC. **Amazon**. 2020. Acesso em: 2 de maio de 2020. Disponível em: <<https://www.amazon.com.br/>>. Citado na página 23.
- AMAZON.COM, INC. **Prime Video**. 2020. Acesso em: 2 de maio de 2020. Disponível em: <<https://www.primevideo.com/>>. Citado na página 23.
- BAY, H.; TUYTELAARS, T.; GOOL, L. V. Surf: Speeded up robust features. In: **Proceedings of the European Conference on Computer Vision**. Berlin, Heidelberg: Springer, 2006. p. 404–417. ISBN 978-3-540-33833-8. Citado na página 29.
- BRADSKI, G. The OpenCV Library. **Dr. Dobb's Journal of Software Tools**, 2000. Citado na página 29.
- BRADSKI, G.; KAEHLER, A. **Learning OpenCV: Computer vision with the OpenCV library**. 1. ed. Sebastopol, CA, USA: O'Reilly, 2008. Citado 2 vezes nas páginas 29 e 30.
- BRAGA, D. F. M. M. d. et al. **Algoritmos de processamento da linguagem natural para sistemas de conversão texto-fala em português**. Tese (Doutorado) — Faculdade de Filologia - Universidade da Coruña, 2008. Disponível em: <<http://hdl.handle.net/2183/1011>>. Citado na página 34.
- CISCO, V. N. I. Cisco visual networking index: Forecast and trends 2017–2022 white paper. **Cisco Internet Report**, v. 17, 2019. Disponível em: <https://www.cisco.com/c/pt_pt/about/press/news-archive-2018/20181127.html>. Citado 2 vezes nas páginas 13 e 23.
- CONCI, A. **Computação Gráfica Teoria e Prática**. 2. ed. Rio de Janeiro, RJ, Brasil: Elsevier, 2008. Citado na página 27.
- CURY, A. **Ansiedade: Como Enfrentar o Mal do Século**. 1. ed. São Paulo, SP, Brasil: Editora Saraiva, 2014. Citado na página 24.
- DADAX. **Internet Live Stat**. 2020. Acesso em: 2 de maio de 2020. Disponível em: <<https://www.internetlivestats.com/>>. Citado na página 23.
- DANELLJAN, M. et al. Accurate scale estimation for robust visual tracking. In: **Proceedings of the British Machine Vision Conference**. Nottingham, United Kingdom: BMVC Press, 2014. Citado 2 vezes nas páginas 44 e 48.
- FISHER, R. A. The use of multiple measurements in taxonomic problems. **Annals of Eugenics**, Wiley Online Library, v. 7, n. 2, p. 179–188, 1936. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-1809.1936.tb02137.x>>. Citado na página 28.
- GONZALEZ, R. E. W. R. C. **Processamento digital de imagens**. 3. ed. São Paulo, SP, Brasil: Prentice Hall, 2009. Citado na página 28.

GUO, J. et al. Fusion of WiFi and vision based on smart devices for indoor localization. In: **Proceedings of the 16th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry**. New York, New York, USA: ACM Press, 2018. (VRCAI '18), p. 1–8. ISBN 9781450360876. Disponível em: <<http://dl.acm.org/citation.cfm?doid=3284398.3284401>>. Citado 2 vezes nas páginas 39 e 41.

GUTIERREZ, B. A. O. et al. Impacto econômico da doença de Alzheimer no Brasil: é possível melhorar a assistência e reduzir custos? **Ciencia e Saude Coletiva**, Fap-Unifesp (SciELO), v. 19, n. 11, p. 4479–4486, 2014. ISSN 16784561. Disponível em: <<https://doi.org/10.1590/1413-812320141911.03562013>>. Citado na página 23.

HAN, J.; KAMBER, M.; PEI, J. **Data Mining : Concepts and Techniques : Concepts and Techniques**. 3. ed. Boston: Elsevier, 2012. Citado na página 31.

HARRIS, C. R. et al. Array programming with NumPy. **Nature**, Springer Science and Business Media LLC, v. 585, n. 7825, p. 357–362, set. 2020. Disponível em: <<https://doi.org/10.1038/s41586-020-2649-2>>. Citado na página 30.

HE, R. et al. Efficient image retrieval based mobile indoor localization. In: **2015 Visual Communications and Image Processing, VCIP 2015**. Singapore: IEEE, 2015. v. 13, p. 1–4. ISBN 9781467373142. Citado 2 vezes nas páginas 38 e 41.

IEEE. Video google: A text retrieval approach to object matching in videos. In: **Proceedings of the IEEE International Conference on Computer Vision**. Sydney, Australia, 2003. p. 1470–1477. Citado na página 33.

JURAFSKY, D.; MARTIN, J. H. **Speech and Language Processing: An Introduction to Natural Language Processing**. 2. ed. [S.l.]: Prentice Hall, 2008. ISBN 0131873210. Citado 2 vezes nas páginas 34 e 35.

KABBAI, L.; ABDELLAOUI, M.; DOUIK, A. Image classification by combining local and global features. **The Visual Computer**, Springer Nature, v. 35, n. 5, p. 679–693, may 2019. ISSN 01782789. Disponível em: <<http://link.springer.com/10.1007/s00371-018-1503-0>>. Citado 3 vezes nas páginas 32, 40 e 41.

KING, D. E. Dlib-ml: A machine learning toolkit. **Journal of Machine Learning Research**, v. 10, p. 1755–1758, 2009. Disponível em: <<https://jmlr.csail.mit.edu/papers/volume10/king09a/king09a.pdf>>. Citado na página 31.

KRISHNA, R. **Computer Vision: Foundations and Applications**. Stanford University, 2017. Disponível em: <http://vision.stanford.edu/teaching/cs131_fall1718/files/cs131-class-notes.pdf>. Citado na página 27.

LEUTENEGGER, S.; CHLI, M.; SIEGWART, R. Brisk: Binary robust invariant scalable keypoints. In: **Proceedings of the IEEE International Conference on Computer Vision**. Barcelona, Espanha: IEEE, 2011. p. 2548–2555. Citado na página 29.

LIU, Y.; ZHANG, H. Indexing visual features: Real-time loop closure detection using a tree structure. In: **Proceedings of the IEEE International Conference on Robotics and Automation**. Barcelona: IEEE, 2012. p. 3613–3618. ISBN 9781467314039. ISSN 10504729. Citado 2 vezes nas páginas 40 e 41.

LOWE, D. G. Distinctive image features from scale-invariant keypoints. **International Journal of Computer Vision**, Springer, v. 60, n. 2, p. 91–110, nov 2004. ISSN 09205691. Disponível em: <<http://link.springer.com/10.1023/B:VISI.0000029664.99615.94>>. Citado 3 vezes nas páginas 23, 28 e 32.

LYRA, G. **OpenVT: Aplicação Escalável de Código Aberto para Busca de Imagens Semelhantes Utilizando Árvore de Vocabulário**. Trabalho de Conclusão de Curso — Universidade Federal do Pampa, jun 2018. Disponível em: <<http://dspace.unipampa.edu.br:8080/jspui/handle/rii/3330>>. Citado 5 vezes nas páginas 32, 33, 34, 41 e 45.

MASLACH, C.; LEITER, M. P. Burnout. In: FINK, G. (Ed.). **Stress: Concepts, Cognition, Emotion, and Behavior: Handbook of Stress**. San Diego: Elsevier, 2016. p. 351–357. ISBN 9780128009512. Disponível em: <<http://www.sciencedirect.com/science/article/pii/B9780128009512000443>>. Citado na página 24.

MCKINNEY, W. Data structures for statistical computing in python. In: WALT, S. van der; MILLMAN, J. (Ed.). **Proceedings of the Python in Science Conference**. Austin, TX, USA: [s.n.], 2010. v. 9, p. 56 – 61. Disponível em: <<http://conference.scipy.org/proceedings/scipy2010/pdfs/mckinney.pdf>>. Citado na página 31.

MIGHALI, V. et al. Innovative IoT-aware services for a smart museum. In: **Proceedings of the 24th International Conference on World Wide Web**. ACM Press, 2015. p. 547–550. Disponível em: <<https://doi.org/10.1145/2740908.2744711>>. Citado 2 vezes nas páginas 39 e 41.

NETFLIX INC. **Netflix**. 2020. Acesso em: 2 de maio de 2020. Disponível em: <<https://www.netflix.com/br/>>. Citado na página 23.

NISTER, D.; STEWENIUS, H. Scalable recognition with a vocabulary tree. In: **IEEE Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition**. New York, NY, USA, 2006. v. 2, p. 2161–2168. Disponível em: <<https://www.cse.unr.edu/~bebis/CS491Y/Papers/Nister06.pdf>>. Citado 2 vezes nas páginas 33 e 45.

PADILHA, V. A.; CARVALHO, A. C. P. de Leon Ferreira de. **Mineração de Dados em Python**. 1. ed. São Carlos, SP, Brasil: ICMC-USP, 2017. Disponível em: <https://edisciplinas.usp.br/pluginfile.php/4109668/mod_resource/content/2/mineracaodadosbiologicos-parte4-completo.pdf>. Citado na página 31.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011. Disponível em: <<https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html>>. Citado na página 31.

PETERSEN, K. et al. Systematic Mapping Studies in Software Engineering. In: **Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering**. Swindon, United Kingdom: BCS Learning & Development Ltd., 2008. v. 8, p. 68–77. Disponível em: <<https://scienceopen.com/document?vid=6d552894-2cc3-4e2b-a483-41fa48a37ef8>>. Citado na página 37.

PETRAITIS, T. et al. Environment recognition based on images using bag-of-words. In: **Proceedings of the 9th International Joint Conference on Computational Intelligence**. SCITEPRESS - Science and Technology Publications, 2017. p. 166–176. Disponível em: <<https://www.scitepress.org/Papers/2017/65856/65856.pdf>>. Citado 2 vezes nas páginas 40 e 41.

ROSSUM, G. **Python Reference Manual**. Amsterdam, The Netherlands, The Netherlands, 1995. Citado na página 30.

RUBLEE, E. et al. ORB: An efficient alternative to SIFT or SURF. In: **CITeseer. Proceedings of the IEEE International Conference on Computer Vision**. IEEE, 2011. v. 11, n. 1, p. 2564–2571. ISBN 9781457711015. Disponível em: <<http://ieeexplore.ieee.org/document/6126544/>>. Citado na página 29.

RUSSELL, S. J.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. 3. ed. Malaysia: Pearson, 2009. Citado na página 35.

SARAIVA E SICILIANO S.A. **Livraria Saraiva**. 2020. Acesso em: 2 de maio de 2020. Disponível em: <<https://www.saraiva.com.br/>>. Citado na página 23.

SERRANO, N.; SAVAKIS, A.; LUO, J. A computationally efficient approach to indoor/outdoor scene classification. In: **Proceedings - International Conference on Pattern Recognition**. IEEE, 2002. v. 16, p. 146–149. Disponível em: <<https://doi.org/10.1109/icpr.2002.1047420>>. Citado 2 vezes nas páginas 38 e 41.

SHAPIRO, L.; STOCKMAN, G. **Computer Vision**. 1. ed. Los Angeles, CA, USA: Prentice Hall, 2001. Citado 2 vezes nas páginas 27 e 28.

SIVAN, S.; DARSAN, G. Computer vision based assistive technology for blind and visually impaired people. In: **ACM International Conference Proceeding Series**. New York, NY, USA: ACM, 2016. v. 7, p. 1–8. ISBN 9781450341790. Disponível em: <<http://doi.acm.org/10.1145/2967878.2967923>>. Citado 3 vezes nas páginas 39, 41 e 42.

SIVIC, J.; ZISSERMAN, A. Video google: A text retrieval approach to object matching in videos. In: **Proceedings of the IEEE International Conference on Computer Vision**. IEEE, 2003. v. 2, p. 1470–1477. ISBN 0-7695-1950-4. ISSN 15505499. Disponível em: <<http://ieeexplore.ieee.org/document/1238663/>>. Citado na página 23.

SUGANTHI S. HANUMANTHAPPA, M. R. L. Classification of event image set using mining techniques. **Indian Journal of Science and Technology**, Indian Society for Education and Environment, v. 9, n. 22, jun 2016. ISSN 09745645. Disponível em: <<https://indjst.org/articles/classification-of-event-image-set-using-mining-techniques>>. Citado 2 vezes nas páginas 39 e 41.

SZELISKI, R. **Computer Vision**. London, United Kingdom: Springer, 2011. (Texts in Computer Science). Citado na página 28.

TAREEN, S. A. K.; SALEEM, Z. A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK. In: **Proceedings of the 2018 International Conference on Computing, Mathematics and Engineering Technologies: Invent, Innovate and Integrate for Socioeconomic Development**. IEEE,

2018. v. 2018-January, p. 1–10. ISBN 9781538613702. Disponível em: <<https://doi.org/10.1109/icomet.2018.8346440>>. Citado na página 41.

ZHOU, W.; LI, H.; FELLOW, T. I. Recent advance in content-based image retrieval: A literature survey. **arXiv**, abs/1706.06064, 2017. Disponível em: <<http://arxiv.org/abs/1706.06064>>. Citado na página 23.

Apêndices

APÊNDICE A – TEMPOS E SUCESSOS DAS INSERÇÕES

Tabela 3 – Tabela dos tempos das dez inserções de cada objeto na gaveta

Livro	Kindle		Remédio		Baralho		Controle	
	Tempo(s)							
21,27	Sucesso							
26,855	Sucesso	Fracasso	22,109	Sucesso	24,626	Sucesso	18,098	Sucesso
19,558	Sucesso	21,343	19,845	Sucesso	24,422	Sucesso	20,412	Sucesso
22,888	Sucesso	24,397	22,457	Sucesso	16,942	Sucesso	29,248	Sucesso
22,71	Sucesso	20,247	23,27	Sucesso	27,617	Sucesso	21,103	Sucesso
17,047	Sucesso	19,386	33,517	Sucesso	16,346	Sucesso	27,253	Sucesso
23,828	Sucesso	22,029	Fracasso	Fracasso	20,003	Sucesso	22,497	Sucesso
17,199	Sucesso	Fracasso	23,423	Sucesso	15,284	Sucesso	18,425	Sucesso
23,307	Sucesso	20,92	27,548	Sucesso	19,466	Sucesso	16,344	Sucesso
Fracasso	Sucesso	37,512	25,137	Sucesso	20,759	Sucesso	31,066	Sucesso
	Fracasso	19,425	21,448	Sucesso	16,921	Sucesso	17,778	Sucesso

Fonte: Próprio Autor

Tabela 4 – Tabela dos tempos das dez inserções de cada objeto na prateleira

Livro	Kindle		Remédio		Baralho		Controle	
	Tempo(s)	Sucesso	Tempo(s)	Sucesso	Tempo(s)	Sucesso	Tempo(s)	Sucesso
25,500	19,074	Sucesso	15,300	Sucesso	20,552	Sucesso	20,989	Sucesso
22,632	Fracasso	Sucesso	19,809	Sucesso	13,141	Sucesso	16,353	Sucesso
18,676	Fracasso	Sucesso	17,380	Sucesso	17,734	Sucesso	14,015	Sucesso
17,316	23,890	Sucesso	18,539	Sucesso	14,965	Sucesso	22,225	Sucesso
27,963	18,224	Sucesso	16,523	Sucesso	18,347	Sucesso	16,912	Sucesso
22,077	21,021	Sucesso	11,517	Sucesso	15,757	Sucesso	21,253	Sucesso
23,309	15,756	Sucesso	17,239	Sucesso	14,194	Sucesso	18,348	Sucesso
24,287	16,701	Sucesso	14,846	Sucesso	13,912	Sucesso	20,762	Sucesso
29,170	18,520	Sucesso	18,552	Sucesso	15,516	Sucesso	Fracasso	Fracasso
36,075	17,295	Sucesso	19,931	Sucesso	14,176	Sucesso	20,059	Sucesso

Fonte: Próprio Autor

Tabela 5 – Tabela dos tempos das dez inserções de cada objeto no roupeiro

Livro	Kindle		Remédio		Baralho		Controle	
	Tempo(s)	Sucesso	Tempo(s)	Sucesso	Tempo(s)	Sucesso	Tempo(s)	Sucesso
21,144	13,364	Sucesso	24,173	Sucesso	35,744	Sucesso	17,134	Sucesso
24,186	21,130	Sucesso	14,712	Sucesso	21,108	Sucesso	15,633	Sucesso
23,256	20,617	Sucesso	17,007	Sucesso	15,596	Sucesso	17,382	Sucesso
19,935	19,980	Sucesso	21,038	Sucesso	22,139	Sucesso	15,879	Sucesso
17,247	21,285	Sucesso	Fracasso	Fracasso	29,844	Sucesso	16,308	Sucesso
20,411	18,441	Sucesso	20,734	Sucesso	19,424	Sucesso	14,328	Sucesso
18,473	20,453	Sucesso	32,963	Sucesso	22,598	Sucesso	25,370	Sucesso
19,101	25,827	Sucesso	18,469	Sucesso	23,850	Sucesso	13,279	Sucesso
19,647	19,732	Fracasso	25,370	Sucesso	36,313	Fracasso	15,974	Sucesso
23,544	19,306	Fracasso	46,930	Sucesso	15,822	Sucesso	14,226	Sucesso

Fonte: Próprio Autor

Tabela 6 – Tabela dos tempos das dez inserções de cada objeto na caixa

Livro		Kindle		Remédio		Baralho		Controle	
Tempo(s)	Sucesso								
18,649	Sucesso	16,125	Sucesso	17,975	Sucesso	17,094	Sucesso	12,963	Sucesso
13,579	Sucesso	12,521	Sucesso	14,282	Sucesso	12,199	Sucesso	13,668	Sucesso
16,369	Sucesso	11,893	Sucesso	18,061	Sucesso	17,158	Sucesso	16,720	Sucesso
14,012	Sucesso	17,746	Sucesso	16,667	Sucesso	17,937	Sucesso	16,549	Sucesso
13,867	Sucesso	14,398	Sucesso	13,883	Sucesso	21,548	Sucesso	16,188	Sucesso
18,794	Sucesso	14,777	Sucesso	20,235	Sucesso	15,186	Sucesso	16,839	Sucesso
17,967	Sucesso	14,346	Sucesso	14,648	Sucesso	16,768	Sucesso	14,700	Sucesso
19,117	Sucesso	15,529	Sucesso	15,230	Sucesso	17,007	Sucesso	15,618	Sucesso
17,793	Sucesso	15,067	Sucesso	14,563	Sucesso	13,236	Sucesso	15,438	Sucesso
20,789	Sucesso	17,043	Sucesso	16,787	Sucesso	19,366	Sucesso	19,382	Sucesso

Fonte: Próprio Autor

APÊNDICE B – TEMPOS E QUALIDADES DAS CONSULTAS TEXTUAIS

Tabela 7 – Tabela das qualidades e tempos das dez consultas de cada objeto na gaveta

Tempo(s)	Livro		Kindle		Remédio		Baralho		Controle	
	Qualidade	Tempo(s)	Qualidade	Tempo(s)	Qualidade	Tempo(s)	Qualidade	Tempo(s)	Qualidade	Tempo(s)
0,154	Excelente	0,088	Inconclusivo	0,106	Bom	0,118	Excelente	0,100	Inconclusivo	
0,094	Excelente	0,076	Excelente	0,091	Bom	0,087	Excelente	0,072	Excelente	
0,067	Excelente	0,093	Excelente	0,102	Excelente	0,118	Bom	0,077	Aceitável	
0,108	Excelente	0,134	Excelente	0,092	Excelente	0,092	Aceitável	0,088	Aceitável	
0,058	Excelente	0,084	Inconclusivo	0,106	Excelente	0,083	Bom	0,076	Aceitável	
0,059	Excelente	0,095	Inconclusivo	0,125	Inconclusivo	0,095	Excelente	0,070	Excelente	
0,058	Excelente	0,087	Inconclusivo	0,107	Excelente	0,088	Excelente	0,082	Aceitável	
0,093	Excelente	0,085	Excelente	0,087	Excelente	0,089	Excelente	0,082	Excelente	
0,051	Inconclusivo	0,113	Bom	0,085	Excelente	0,094	Excelente	0,083	Excelente	
0,095	Inconclusivo	0,134	Bom	0,093	Excelente	0,072	Excelente	0,069	Insatisfatório	

Fonte: Próprio Autor

Tabela 8 – Tabela das qualidades e tempos das dez consultas de cada objeto na prateleira

Tempo(s)	Livro	Kindle		Remédio		Baralho		Controle		
		Qualidade	Tempo(s)	Qualidade	Tempo(s)	Qualidade	Tempo(s)	Qualidade	Tempo(s)	
0,084		Bom	0,427	Excelente	0,157	Excelente	0,098	Excelente	0,089	Bom
0,073		Excelente	0,092	Inconclusivo	0,139	Excelente	0,132	Aceitável	0,082	Bom
0,089		Excelente	0,093	Inconclusivo	0,104	Excelente	0,216	Excelente	0,077	Bom
0,208		Inconclusivo	0,097	Bom	0,240	Excelente	0,081	Excelente	0,075	Excelente
0,203		Inconclusivo	0,147	Bom	0,133	Excelente	0,055	Bom	0,093	Bom
0,084		Inconclusivo	0,075	Bom	0,116	Bom	0,066	Aceitável	0,082	Bom
0,082		Insatisfatório	0,106	Bom	0,191	Bom	0,088	Bom	0,069	Bom
0,078		Bom	0,075	Excelente	0,112	Bom	0,055	Excelente	0,062	Excelente
0,075		Excelente	0,064	Excelente	0,205	Excelente	0,094	Excelente	0,067	Inconclusivo
0,081		Aceitável	0,058	Excelente	0,067	Excelente	0,076	Excelente	0,068	Bom

Fonte: Próprio Autor

Tabela 9 – Tabela das qualidades e tempos das dez consultas de cada objeto no roupeiro

Tempo(s)	Livro	Kindle		Remédio		Baralho		Controle		
		Qualidade	Tempo(s)	Qualidade	Tempo(s)	Qualidade	Tempo(s)	Qualidade	Tempo(s)	
0,107		Excelente	0,751	Bom	0,092	Inconclusivo	3,659	Bom	0,112	Bom
0,084		Excelente	0,069	Excelente	0,342	Inconclusivo	0,073	Bom	0,067	Bom
0,093		Excelente	0,107	Excelente	0,104	Excelente	0,078	Bom	0,069	Bom
0,227		Excelente	0,085	Excelente	0,090	Excelente	0,091	Bom	0,083	Excelente
0,094		Excelente	0,101	Excelente	0,076	Inconclusivo	0,076	Excelente	0,092	Excelente
0,074		Excelente	0,082	Bom	0,142	Excelente	0,076	Excelente	0,071	Aceitável
0,077		Excelente	0,134	Excelente	0,075	Aceitável	0,119	Inconclusivo	0,073	Excelente
0,089		Bom	0,111	Bom	0,070	Excelente	0,077	Excelente	0,073	Bom
0,090		Inconclusivo	0,083	Bom	0,116	Excelente	0,101	Inconclusivo	0,070	Excelente
0,088		Aceitável	0,333	Excelente	0,076	Excelente	0,136	Excelente	0,294	Excelente

Fonte: Próprio Autor

Tabela 10 – Tabela das qualidades e tempos das dez consultas de cada objeto na caixa

Livro Tempo(s)	Kindle		Remédio		Baralho		Controle		
	Qualidade	Tempo(s)	Qualidade	Tempo(s)	Qualidade	Tempo(s)	Qualidade	Tempo(s)	
0,081	Excelente	0,082	Aceitável	0,073	Inconclusivo	0,077	Excelente	0,084	Bom
0,088	Excelente	0,073	Excelente	0,083	Excelente	0,066	Excelente	0,091	Excelente
0,068	Excelente	0,076	Excelente	0,097	Excelente	0,057	Bom	0,105	Excelente
0,119	Excelente	0,114	Excelente	0,102	Aceitável	0,065	Aceitável	0,080	Bom
0,083	Excelente	0,072	Excelente	0,097	Excelente	0,049	Excelente	0,053	Excelente
0,092	Excelente	0,079	Excelente	0,086	Excelente	0,056	Bom	0,061	Excelente
0,083	Excelente	0,067	Excelente	0,164	Excelente	0,075	Bom	0,086	Excelente
0,063	Aceitável	0,112	Excelente	0,089	Excelente	0,061	Excelente	0,086	Excelente
0,076	Bom	0,066	Excelente	0,092	Excelente	0,051	Excelente	0,065	Excelente
0,079	Excelente	0,080	Excelente	0,069	Bom	0,066	Excelente	0,101	Excelente

Fonte: Próprio Autor