ESTG

INSTITUTO POLITÉCNICO
DE VIANA DO CASTELO

Aplicação móvel com recurso a posicionamento visual para auxiliar na orientação de pessoas invisuais

Aplicação móvel com recurso a posicionamento visual para auxiliar na orientação de pessoas invisuais

2022

Escola Superior de Tecnologia e Gestão

INSTITUTO POLITÉCNICO
DE VIANA DO CASTELO

Joana Torres Gonçalves

APLICAÇÃO MÓVEL COM RECURSO A POSICIONAMENTO VISUAL PARA AUXILIAR NA ORIENTAÇÃO DE PESSOAS INVISUAIS

Mestrado em Engenharia Informática

Trabalho efectuado sob a orientação do
Professora Doutora Sara Paiva

Fevereiro de 2022

Agradecimentos

Agradeço à minha orientadora Sara Paiva por aceitar conduzir o meu trabalho de pesquisa e pelo constante apoio. A ela, agradeço ainda a oportunidade de participar na conferência IEEE International Smart Cities Conference 2021, onde foi submetido um artigo baseado no projeto da aplicação.

Aos meus pais e irmão que estiveram sempre ao meu lado apoiando-me ao longo de toda a minha trajetória.

A todos os meus amigos do curso de mestrado que compartilharam dos inúmeros desafios que enfrentamos, sempre com o espírito colaborativo.

A todos os outros, muitos, a quem devo o suficiente para ter conseguido efetuar a presente dissertação.

Abstract

Blindness is a limitation that constitutes a barrier for social inclusion and which affects the performance of several tasks that have a direct impact in the quality of life of all those who live with this condition. The fast pace of innovation and advances in technological research, has given people with visual impairment hope to find more suitable ways to move in urban environments, enjoy greater independence and thus have a better quality of life. One of the main challenges for people with visual impairments is mobility in urban environments, where attaining a notion of positioning can be extremely useful and a way to contribute to greater autonomy. This project proposes an application with the aim of supporting people who suffer from this impairment, helping their navigation and orientation. The development of the proposed solution resorts to the use of image recognition technologies, allowing the identification of different reference points in order to help the user to obtain guidance on their current location. The proposed application uses Google Cloud Vision, in order to present a new approach with regard to geolocation, which was subjected to individual tests and also integration tests in the city of Braga. After evaluation of the tests carried out, the proposed solutions proved to be a viable option in improving mobility for blind and partially sighted people.

Keywords: Visually Impaired People, Google Cloud Vision, Image Recognition.

Resumo

A cegueira é uma limitação que constitui um bloqueio à participação na sociedade e inclusão social, refletindo-se diariamente na execução de algumas tarefas e numa menor qualidade de vida.

O ritmo acelerado da inovação e dos avanços na investigação tecnológica deu esperança às pessoas portadoras de uma deficiência visual, de encontrar formas mais adequadas de se deslocarem em ambientes urbanos, usufruírem de maior independência e terem assim melhor qualidade de vida.

Um dos principais desafios de pessoas com deficiência visual é a mobilidade em ambientes urbanos, onde obter a noção de posicionamento pode ser extremamente útil e uma forma de contribuir para uma maior autonomia.

Este projeto propõe uma aplicação com o objetivo de servir de suporte a pessoas que sofrem desta limitação, auxiliando na sua navegação e orientação.

O desenvolvimento desta solução recorre ao uso de tecnologias de reconhecimento de imagem, permitindo a identificação de diversos pontos de referência de forma a ajudar o utilizador a obter orientação sobre a sua localização atual.

A aplicação apresentada usa por base o Google Cloud Vision, de forma a apresentar uma nova abordagem no que toca a geolocalizacão, tendo a mesma sido sujeita a testes individuais e mais tarde integradores na cidade de Braga.

Após uma avaliação dos testes realizados, esta aplicação demonstra ser uma opção viável na ajuda da mobilidade para pessoas cegas e amblíopes.

Palavras-chave: Deficiência Visual, Google Cloud Vision, Reconhecimento de Imagem

Conteúdo

Agradecimentos **1**

Siglas e Acrónimos **4**

1 Introdução **8**

1.1 Motivação 8

1.2 Objetivos 10

1.3 Abordagem metodológica 10

1.4 Estrutura do documento 12

2 A Deficiência Visual **13**

2.1 Segmento de pessoas cegas e amblíopes 13

2.2 Formas de Orientação e Mobilidade 15

2.3 Cidades inteligentes e pessoas cegas e amblíopes 16

2.3.1 Tecnologias em cidades inteligentes 17

2.3.2 Sistemas de Transporte em Cidades Inteligentes para deficientes visuais 19

2.3.3 Casas inteligentes 20

3 Enquadramento conceptual **21**

3.1 Sistema operativo Android 21

3.1.1 Acessibilidade 24

3.1.2 Java e Kotlin 24

3.1.3 Android Studio 25

3.2 Pontos de referência 26

3.3 Reconhecimento de imagem 26

3.4 Modelos de Redes Neurais 27
3.4.1 Rede Neural Convolucional 27
3.4.2 Redes Neurais Convolucionais Regionais 28
3.4.3 Fast R-CNN 28
3.5 OCR 29

4 Revisão da literatura **32**
4.1 Navegação de pessoas cegas e amblíopes 32
4.2 Posicionamento exterior usando pontos de referência 37

5 Desenvolvimento da aplicação **39**
5.1 Arquitetura 39
5.2 Escolha da framework de reconhecimento de imagem 40
5.2.1 Google Cloud Vision API 40
5.2.2 IBM Watson Visual Recognition API 41
5.2.3 Clarifai API 42
5.2.4 Microsoft Computer Vision API 42
5.2.5 Amazon Rekognition 43
5.2.6 Comparação das APIs 43
5.3 Desenvolvimento da aplicação móvel 45
5.3.1 Visão geral 45
5.3.2 Configuração da Google Cloud Vision API 46
5.3.3 Layout 48
5.3.4 Integração da câmara 49
5.3.5 Integração da *Google Cloud Vision API* com a aplicação 49
5.3.6 Protótipo final 54

6 Avaliação **56**
6.1 Metodologia 56
6.1.1 Reconhecimento do logotipo 57
6.1.2 Reconhecimento de texto 58
6.1.3 Reconhecimento de pontos de referência 62

6.1.4 Testes em cenário real 64

7 Conclusão e trabalho futuro **67**

Referências **68**

Siglas e Acrónimos

ACAPO Associação dos Cegos e Amblíopes de Portugal

CNN Rede neural convolucional

DSR Design Science Research

GPS Sistema de posicionamento global

OCR Reconhecimento Óptico de Caracteres

OMS Organização Mundial de Saúde

PDR Pedestrian Dead Reckoning

RFID Identificação por radiofrequência

SIG Sistema de informação geográfico

SLAM Localização e Mapeamento Simultâneos

TIC Tecnologias de Informação e Comunicação

Lista de Figuras

1.1 Atividades DSR (adaptado de [51]) 11

2.1 Causas da deficiência visual mundial segundo a Organização Mundial de Saúde (OMS), 2010 (adaptado de [71]) 14
2.2 Técnica do Guia Vidente (Retirado de [62]) 16
2.3 Mapa táctil (Retirado de [45]) 19

3.1 Arquitetura do Sistema Operativo Android (adaptado de [7])) 23
3.2 Deteção de objetos via RCNN (Fonte: [26]) 28
3.3 Exemplo de reconhecimento automático de matrículas em aplicações de gestão de parques de estacionamento (Retirado de [49]) 30

5.1 Diagrama de arquitetura 40
5.2 Activação do Google Cloud API 47
5.3 Criação da chave 47
5.4 Dependências 47
5.5 Permissão à Internet 48
5.6 Configuração da classe *Vision Builder* 48
5.7 Layout 49
5.8 Função setupCameraPreview() 50
5.9 Reconhecimento de Logotipo 54
5.10 Reconhecimento de um ponto de referência 55
5.11 Reconhecimento de texto 55

6.1 Logótipos usados para o reconhecimento 57
6.2 Avaliação dos resultados obtidos nos logótipos 58

6.3 Imagens usadas para a avaliação do reconhecimento de texto. Da esquerda para a direita: (a) MAMMA MIA Ristorante Pizzeria; (b) 4710-079 Rua José Antunes Guimarães Gualtar; (c) ARRANJOS DE ROUPA D.AMÉLIA 59
6.4 Formula da precisão usada para reconhecimento de texto 59
6.5 Resultados obtidos no ângulo frontal 60
6.6 Resultados de reconhecimento de ângulo lateral 61
6.7 Resultados do reconhecimento de texto de 3 metros de distância 62
6.8 Imagens usadas para o reconhecimento de pontos de referência: (a) Bom Jesus, na cidade de Braga, (b) Santa Luzia, na cidade de Viana do Castelo e (c) Avenida Central, na cidade de Braga. 63
6.9 Resultados de reconhecimento de pontos de referência 63
6.10 Cenário do mapa dos testes de campo mostrando os lugares a serem reconhecidos (1 a 4) e os lugares onde o reconhecimento foi feito (5 a 11) 64

Lista de Tabelas

2.1 Número de pessoas com deficiência visual e percentagem correspondente da deficiência global por Região e país da OMS, 2010 [71] 15

4.1 Aplicações exteriores 36
4.2 Aplicações interiores 36
4.3 Aplicações interiores/exteriores 37

5.1 Funcionalidade das APIs 44

6.1 Tempo médio e precisão média de cada logotipo 58
6.2 Tempo médio de processamento e precisão média do reconhecimento de texto de ângulo frontal 59
6.3 Tempo médio de processamento e precisão média do reconhecimento de texto de ângulo lateral 60
6.4 Tempo Médio de Processamento e Precisão Média por Distância 61
6.5 Tempo Médio de Processamento e Precisão Média de Reconhecimento de pontos de referência 63
6.6 Correspondência entre os locais a reconhecer e os locais onde o reconhecimento será feito 65
6.7 Tempo Médio de Processamento e Precisão Média do Reconhecimento de texto em testes no mundo real 65
6.8 Tempo Médio de Processamento e Precisão Média do Reconhecimento de Logotipo em testes no mundo real 65
6.9 Tempo Médio de Processamento e Precisão Média de Reconhecimento de Pontos de Referência no cenário real 66

Capítulo 1

Introdução

Com este projeto pretende-se o desenvolvimento de uma aplicação móvel que dê suporte às pessoas cegas e amblíopes, de modo a que estas se possam orientar melhor numa cidade, particularmente em casos onde tenham perdido a orientação.

Neste primeiro capítulo é dada a conhecer a motivação para a realização deste projeto. A seguir, são apresentados os objetivos delineados, assim como a abordagem metodológica usada ao longo de todo o projeto. Por fim, o capítulo termina com a estrutura do resto do documento.

1.1 Motivação

A OMS, no seu relatório "Deficiência Visual 2010", estimou o número total de pessoas com deficiência visual em cerca de 285 milhões [71]. Para além deste número, estimou também o número de pessoas cegas em 39 milhões e o número de pessoas com baixa visão em cerca de 246 milhões. O termo baixa visão é utilizado para descrever uma perda de acuidade visual enquanto se retém alguma visão. Pode ser aplicado a pessoas com visão que não conseguem ler um jornal a uma distância normal, mesmo com a ajuda de óculos ou lentes de contacto [31]. As pessoas com deficiência visual têm alguma visão, mesmo que em termos clínicos sejam consideradas cegas. Algumas têm apenas uma perceção de luz, mas podem tirar proveito desta capacidade, como localizar uma porta [21].

Segundo a Associação dos Cegos e Amblíopes de Portugal (ACAPO), a definição de cegueira consiste numa perda total ou quase total de visão, ou seja, um grau de visão

abaixo de 0.05 [27].

Em Portugal, os valores reportados em 2001 consistiam numa estimativa de cerca de 160 mil pessoas com deficiência visual [5] enquanto que os valores de 2011 apontavam para valores de 900 mil pessoas com deficiência visual, das quais 28 mil sofriam de cegueira [2].

Este aumento do número de pessoas com deficiência visual deve-se, principalmente, ao facto de que em 2011, pela primeira vez, não foram tidos em consideração somente dados referentes a diagnósticos de incapacidade, mas também dados referente a pessoas com outros tipos de défices visuais, tais como baixa visão. Devido a este facto, não é possível avaliar a evolução do número de casos de pessoas com algum tipo de deficiência visual comparativamente ao ano de 2001.

Um dos grandes desafios para as pessoas cegas e amblíopes a nível de navegação numa cidade inclui a dificuldade de se deslocarem de uma origem para um destino de forma autónoma, conseguir posicionar-se quando perdem a orientação ou mesmo detetarem obstáculos no seu caminho, tais como carros e buracos que dificultem a sua circulação [57].

A navegação exterior, apesar de ser essencial, continua a ser uma tarefa difícil para pessoas cegas e amblíopes. A tecnologia disponível para a navegação de cegos não é suficientemente acessível. Analisadas as principais limitações na navegação exterior de pessoas cegas e amblíopes, as abordagens atualmente seguidas, e também tendo feito o estudo sobre a técnica de posicionamento visual, pretende-se conjugar estas duas realidades para aferir a melhoria ao nível do posicionamento que é possível proporcionar a pessoas cegas e amblíopes.

De forma a colmatar esta necessidade, a presente tese apresenta um caso de estudo na cidade de Braga, onde se desenvolveu um protótipo que, através da utilização de algoritmos de reconhecimento de imagem, informa uma pessoa cega e amblíope acerca do seu posicionamento atual. O objetivo final é facilitar a mobilidade das pessoas cegas e amblíopes e promover a sua independência.

1.2 Objetivos

Este projeto tem como principal objetivo o desenvolvimento de uma aplicação móvel com o intuito de ajudar as pessoas cegas e amblíopes a posicionarem-se numa cidade, quando se encontram perdidas. Para tal, é usada a técnica de posicionamento visual, de forma a informar o utilizador cego e amblíope de onde está, relativamente a um ponto conhecido. A aplicação móvel usa algoritmos de reconhecimento de imagem de forma a conseguir inferir a posição e informar o utilizador via áudio através do *TalkBack*.

A aplicação deve cumprir os seguintes requisitos:

- Suportar o reconhecimento de texto: Com a tecnologia Reconhecimento Óptico de Caracteres (OCR), é possível transformar uma imagem num conteúdo legível ao que estava na fotografia inicial. Isto pode ser relevante, pois podemos tirar uma fotografia a um nome de uma loja e assim ser devolvido um resultado que pode ser muito pertinente para a pessoa cega e amblíope para se localizar.

- Suportar o reconhecimento de pontos de referência conhecidos: Esta funcionalidade é útil, pois é possível detetar estruturas famosas, sejam elas naturais, ou construídas pelo homem, numa imagem. O facto de não ser necessário ter que se recorrer a um dataset próprio, para por exemplo, reconhecer o Bom Jesus em Braga, acaba por ser uma funcionalidade útil para os objetivos delineados.

- Suportar o reconhecimento de logótipos: Esta funcionalidade é importante e complementar ao reconhecimento de texto ou ponto de referência. Locais como McDonalds ou Burger King, são locais que podem dar ao utilizador a noção de orientação quando informado da sua proximidade.

1.3 Abordagem metodológica

Para o desenvolvimento deste projeto, a metodologia usada foi o Design Science Research (DSR). Esta enfatiza a conceção e construção de artefactos relevantes, tais como sistemas, aplicações, métodos, e outros, que poderiam potencialmente contribuir para a eficácia dos sistemas de informação nas organizações [51].

Em [52], o autor divide o processo em seis etapas, sendo elas as seguintes:

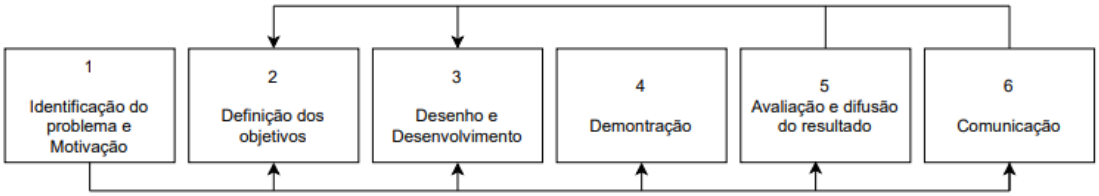

Figura 1.1: Atividades DSR (adaptado de [51])

1. **Identificação do problema e motivação** - Nesta etapa será necessário um bom entendimento do estado de arte na área em questão e definir claramente o aspeto diferenciador ao nível da inovação.

2. **Definição dos objetivos para a solução** - Depois de identificarmos e conhecermos o problema, terá que se definir os objetivos para a solução. Os objetivos tanto podem ser quantitativos como qualitativos. Se forem objetivos quantitativos, estes devem ser melhor que os já existentes, isto é, devem-se definir métricas para permitir uma comparação entre o antes e depois da aplicação da solução. Se forem objetivos qualitativos, a solução deve descrever a forma como os artefactos a serem desenvolvidos permitirão a solução do problema. Nesta etapa, é essencial conhecer outras soluções, caso existam e saber os prós e contras de modo a servir de termo de comparação.

3. **Desenho e desenvolvimento** - é necessário desenhar o artefacto, sendo importante determinar as funcionalidades desejadas do artefacto bem como a sua arquitetura.

4. **Demonstração da solução** - Deve demonstrar-se o funcionamento da aplicação, sendo que pode envolver apenas simulação ou a experiência real do utilizador. Este ponto poderá ser usado para o desenvolvimento de testes de forma a garantir que todo o processo criado é o melhor para responder à questão do problema em causa.

5. **Avaliação** - Na fase de avaliação deveremos verificar se realmente a aplicação suporta a solução para o problema., isto é, determinar se os resultados atingidos vão ao encontro com os resultados esperados e perceber se se adequam à realidade.

6. **Comunicação e difusão do resultado** - Nesta última fase é importante divulgar o resultado final, podendo estes serem publicadas em revista ou conferências.

Estas atividades podem ser executadas pela ordem de 1-6, ou até podem seguir ordens diferentes, e algumas destas atividades serem executadas várias vezes.

1.4 Estrutura do documento

Este documento está estruturado em 7 capítulos. No capítulo 1 (Introdução) é apresentada a motivação, os objetivos deste projeto e a abordagem metodológica usada.

No capítulo 2 (A Deficiência Visual) é destacada a definicão de cegueira, assim como as causas da deficiência visual. Noutro âmbito, são descritas as formas de Orientação e Mobilidade e as tecnologias que são usadas em cidades inteligentes.

No capítulo 3 (Enquadramento Conceptual) são apresentadas as tecnologias relevantes ao desenvolvimento deste projeto e uma breve explicação sobre pontos de referência, reconhecimento de imagem, modelos de redes neuronais e ainda os diversos usos da tecnologia OCR.

No capítulo 4 (Revisão da Literatura) destacam-se aplicações que abordam a problemática da navegação para pessoas cegas e amblíopes existentes no mercado.

O capítulo 5 (Desenvolvimento da aplicação) centra-se na arquitetura utilizada para a abordagem do problema, tal como na escolha da framework utilizada para o reconhecimento de imagem e o desenvolvimento da aplicação móvel.

O capítulo 6 (Avaliação) corresponde aos testes realizados e à respetiva análise dos resultados obtidos.

No capítulo 7 (Conclusão e trabalho futuro) são apresentadas as conclusões face aos resultados obtidos e sugestões de trabalho futuro para melhoramento da aplicação.

Capítulo 2

A Deficiência Visual

2.1 Segmento de pessoas cegas e amblíopes

A cegueira é definida pela OMS como a acuidade visual menor do que 3/60 no melhor olho com a melhor correção óptica [67]. A cegueira, de acordo com o momento da perda visual, pode ser denominada de cegueira congénita, quando essa perda se dá antes dos cinco anos de idade, e cegueira adquirida, após essa idade. Tanto a cegueira congénita quanto a adquirida apresentam etiologias variadas, envolvendo desde questões genéticas e doenças infecciosas a traumas de ordens diversas.

Segundo a OMS, existem duas categorias representativas da deficiência visual, sendo elas a cegueira e a baixa visão, que afetam 285 milhões de pessoas, em que 39 milhões são referentes à cegueira e os restantes 246 milhões a baixa visão.

Podemos então distinguir sete classes de acuidade visual, que são as seguintes:

- Cegueira Total (não possui perceção de luz)
- Próximo à cegueira (menor que 20/1000)
- Baixa visão profunda (20/500 a 20/1000pés)
- Baixa visão severa (20/200 a 20/400pés)
- Baixa visão moderada (20/80 a 20/150pés)
- Próxima do normal (20/30 a 20/60pés)

- Normal (20/12 a 20/25pés)

Para além dos erros refrativos que não são devidamente tratados, existem ainda doenças indicadas no gráfico 2.1, como as cataratas, o glaucoma, o tracoma, a opacidade da córnea, a degeneração macular relacionada com a idade, a retinopatia diabética que podem afetar gravemente a acuidade visual, sendo estas as principais causas, segundo a OMS.

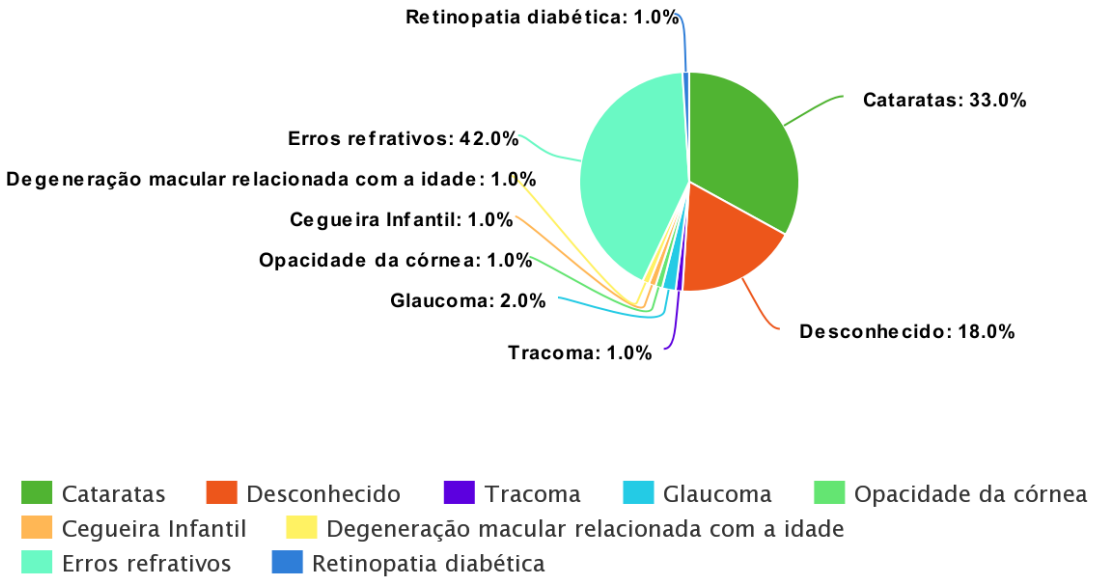

Figura 2.1: Causas da deficiência visual mundial segundo a OMS, 2010 (adaptado de [71])

A OMS salienta que mais de 3/4 dos casos da incapacidade visual podem ou poderiam prevenir-se. A maioria das pessoas com deficiência visual e cegueira tem mais de 50 anos de idade; contudo, a perda da visão pode afetar pessoas de todas as idades.

Na tabela 2.1 está representada a distribuição de pessoas com deficiência visual nas seis regiões da Organização Mundial de Saúde, incluindo a percentagem da deficiência global entre parênteses no ano de 2010.

Cerca de 246 milhões de pessoas em todo o mundo sofrem de baixa visão e 39 milhões são cegas.

Em Portugal, os censos de 2001 [5] existiam cerca de 160 mil pessoas com deficiência visual, enquanto os censos de 2011 [2] mostraram existirem 900 mil pessoas com deficiência visual, das quais 28 mil eram cegas.

O principal motivo para o aumento do número de deficientes visuais é que, pela primeira

		Cegueira	Baixa Visão	Deficiência Visual
Região	Total da população (Milhões)	Número em milhões (percentagem)	Número em milhões (percentagem)	Número em milhões (percentagem)
África	804.9	5.888 (15%)	20.407 (8.3%)	26.295 (9.2%)
Europa	889.2	2.713 (7%)	25.502 (10.4%)	28.215 (9.9%)
América	915.4	3.211 (8%)	23.401 (9.5%)	26.612 (9.3%)
India	1181.4	8.075 (20.5%)	54.544 (22.2%)	62.619 (21.9%)
China	1344.9	8.248 (20.9%)	67.264 (27.3%)	75.512 (26.5%)
Sudeste Asiático (India excluído)	579.1	3.974 (10.1%)	23.938 (9.7%)	27.913 (9.8%)
Oeste do pacífico (China excluído)	1344.9	2.338 (6%)	12.386 (5%)	14.724 (5.2%)
Leste do mediterrâneo	580.2	4.918 (12.5%)	18.581 (7.6%)	23.499 (8.2%)
Total Mundo	6737.5	39.365 (100%)	246.024 (100%)	285.389 (100%)

Tabela 2.1: Número de pessoas com deficiência visual e percentagem correspondente da deficiência global por Região e país da OMS, 2010 [71]

vez em 2011, foram considerados não só os dados de diagnóstico de incapacidade mas também os dados de outros tipos de pessoas com outros tipos de défices visuais. Por isso, não é possível avaliar a evolução do número de pessoas com determinados tipos de deficiência visual em relação ao ano de 2001.

Verifica-se também que a taxa de incidência mais elevada era a da deficiência visual representando 1,6% do total de população, com a mesma proporção entre homens e mulheres.

Após o enquadramento sobre as pessoas cegas e amblíopes e sobre a percentagem de pessoas no mundo invisuais, a próxima secção foca-se num enquadramento mais teórico, acerca das formas de Orientação e Mobilidade.

2.2 Formas de Orientação e Mobilidade

Segundo a ACAPO [3], as pessoas indicadas para o ensino de Orientação e Mobilidade são os profissionais devidamente habilitados para o efeito, uma vez que esse ensino não se limita às técnicas de utilização de guias ou desenvolvimento da capacidade e habilidade de utilizar bengalas. Também é fulcral o desenvolvimento da mobilidade corporal, educação sensorial e o desenvolvimento de determinados conceitos.

As técnicas mais comuns de orientação e mobilidade são: do guia vidente / humano; autoproteção/ autoajuda; bengala; e cão guia.

A técnica do guia vidente é uma forma dependente de locomoção e é utilizada quando por algum motivo, a pessoa não pode utilizar a bengala ou cão-guia, ou quando é mais

apropriado ou recomendado deslocar-se com um guia, em locais com uma grande concentração de pessoas. Através do toque, a pessoa com deficiência visual consegue perceber o movimento do corpo do guia vidente, isto é, se subiu, desceu, virou ou desviou, e com base nisto, participar nas decisões do que ocorre durante o seu deslocamento.

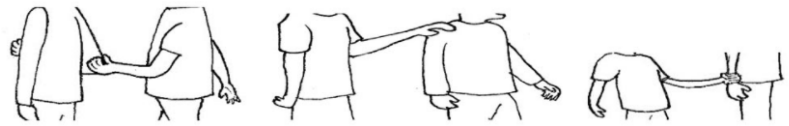

Figura 2.2: Técnica do Guia Vidente (Retirado de [62])

As técnicas de autoajuda / autoproteção são técnicas que possibilitam a pessoa cega e amblíope a movimentar-se com independência e segurança em ambientes desconhecidos ou que ofereçam algum perigo. É composta por proteção de parte superior e parte inferior do corpo.

A técnica da bengala ainda é considerado a técnica de locomoção mais competente. A bengala cano-longo é usada para aumentar a distância da sensação de toque do utilizador. Geralmente é usada com ela para baixo em um movimento de balanço, através do caminho planeado, para detetar obstáculos. No entanto, as técnicas para uso da bengala podem variar dependendo do utilizador e/ou da situação. Algumas pessoas com deficiência visual não possuem esses tipos de bengalas, optando pela menor e mais leve bengala de identificação.

O cão guia representa outro recurso de orientação e mobilidade e faz o trabalho de reconhecer e desviar dos obstáculos, sendo direcionado pela pessoa com deficiência visual, a qual deve orientar-se espacialmente e ter noção sobre qual caminho tomar. Exige do seu utilizador, conhecimentos prévios de Orientação e Mobilidade e condições para a realização dos cuidados e manutenção da sobrevivência, saúde e higiene do cão. São animais treinados especificamente para ajudar pessoas cegas e amblíopes a deslocarem-se em segurança.

2.3 Cidades inteligentes e pessoas cegas e amblíopes

Um dos principais e mais discutidos conceitos, em termos interação da pessoa invisual com o ambiente urbano, é o conceito de cidade inteligente. Uma cidade inteligente é uma

área metropolitana onde se tenha investido em infraestruturas de informação de tal modo a gerar uma melhor qualidade de vida para os cidadãos. Estas infraestruturas incluem, por exemplo, dispositivos que armazenam dados sobre os cidadãos e os processam de forma a melhor gerir sistemas de tráfego e transporte, redes de abastecimento de água, escolas, bibliotecas, hospitais, entre outros [46].

Nas subsecções seguintes irão ser apresentadas algumas soluções existentes para ajudar as pessoas cegas e amblíopes e também mostrar como uma cidade inteligente poderia melhorar essas soluções.

2.3.1 Tecnologias em cidades inteligentes

Hoje em dia, algum dos desafios enfrentados pelas pessoas cegas e amblíopes focam-se na mobilidade e na navegação através de obstáculos conhecidos e desconhecidos. As soluções Tecnologias de Informação e Comunicação (TIC) podem ajudar a mitigar os desafios acima mencionados, fornecendo soluções como:

- Aplicações móveis adaptadas a utilizadores com deficiência visual
- Sistemas de sinais sonoros e vibrotáteis baseados em sistemas de realidade aumentada para utilizadores, que fornecem informações precisas sobre a sua localização.

Aplicações móveis tais como *Seeing Eye GPS* [61] ou *Blind Square* [13] são aplicações de navegação que fazem o uso de *Sistema de posicionamento global (GPS)*. O utilizador introduz o destino pretendido através do comando ou com a ajuda da função *VoiceOver* que está disponível em maior parte dos aparelhos. Os sinais GPS pode indicar às pessoas cegas e amblíopes a sua localização, calcular rotas e transmitir direções através de sinais sonoros ou vibração.

Estas aplicações ajudam as pessoas cegas e amblíopes a deslocarem-se da posição A para B. No entanto, em espaços fechados, tal como dentro de comboios ou dentro de *Shoppings* o sinal torna-se mais fraco, o que torna a navegação mais complicada. Aqui é onde as cidades inteligentes podem revolucionar, com o uso de células pequenas [1] que podem ser usadas para impulsionar a força do sinal.

Outra forma que pode ser útil para as pessoas cegas e amblíopes, é o facto de as cidades inteligentes terem a possibilidade de equipar os edificios com *Beacons*.

Na subsecção seguinte é mostrado como pode ser útil para as pessoas cegas e amblíopes.

Beacons

Os *Beacons* são pequenos transmissores colocados em redor dos edifícios que enviam informações do local em tempo real diretamente para dispositivos móveis. Podem ser instalados em edifícios públicos, escritórios ou pequenos locais como paragens de autocarros [54]. Estes funcionam tanto no interior como no exterior e as pessoas cegas e amblíopes podem ser notificadas através da vibração ou som desde o seu dispositivo móvel.

Um exemplo de onde podemos ver o uso dos *Beacons* é na cidade de Varsóvia, Polónia, onde foi desenvolvido uma rede de milhares de *Beacons* para ajudar os deficientes invisuais a deslocarem-se na cidade [23].

Outra aplicação que podemos encontrar é a *Wayfindr* [70] que envia ao utilizador informações sobre a sua proximidade através de instruções áudio.

Por outro lado, as cidades inteligentes também podem fornecer estruturas urbanas inteligentes ou chão tátil que facilita a navegação de pessoas cegas e amblíopes que irão ser falados na subsecção seguinte.

Mapas de cidades tacteis e falantes

Cidades inteligentes também disponibilizam estruturas urbanas inteligentes ou chão táctil que facilitam a navegação das pessoas com défice visual. O pavimento táctil compensa a ausência de meio-fio dos passeios e as pessoas invisuais podem utilizar a informação debaixo dos pés.

Os mapas físicos servem como uma adição às aplicações de posicionamento e *Beacons*. Enquanto os dispositivos móveis podem ajudar alguém a navegar no local, os mapas físicos têm a escala para fornecer uma visão global de um espaço.

Os mapas de cidade tácteis são fáceis de ler e podem oferecer detalhes úteis sobre distâncias, estrutura de edifícios, gradientes de ruas e outras características topográficas.

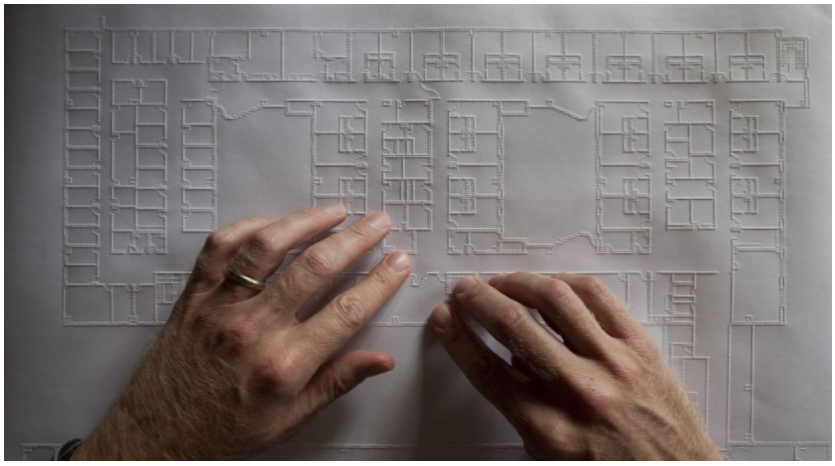

Figura 2.3: Mapa táctil (Retirado de [45])

2.3.2 Sistemas de Transporte em Cidades Inteligentes para deficientes visuais

Apanhar um autocarro ou um comboio pode ser uma tarefa trivial que a maioria de nós realiza diariamente, sobretudo nas grandes cidades, onde a maioria das pessoas utiliza o transporte público como meio de transporte principal. As pessoas com deficiência visual enfrentam muitos desafios em aceder ao sistema de transportes públicos. Os transportes públicos têm de ser concebidos de forma a ajudar a reduzir a dificuldade enfrentada pelas pessoas com deficiência visual.

A aplicação de nome *GeorgiePhone* [53] desenvolvida no Reino Unido destina-se principalmente a pessoas que sofrem de baixa visão. A aplicação consiste num sistema de localização de autocarros, entre outras funcionalidades, para localizar a paragem de autocarro mais próxima, ler em voz alta o nome da paragem de autocarro cada vez que o autocarro para e indica horários de chegada do autocarro.

No Canadá, uma aplicação chamada iBus, baseada num projeto centrado no cliente, utiliza GPS e odómetros para dar informação em tempo real sobre todos os autocarros da cidade de Montreal, permitindo a todos os citadinos, incluindo os com todo o tipo de deficiências, ter a forma mais fácil de utilizar o transporte público. Com esta aplicação, os motoristas de autocarro podem mudar de itinerário conforme o tráfego ou os passageiros, e se isso ocorrer os passageiros são alertados no aplicativo, no interior do painel do autocarro e também os autocarros comunicam com as luzes de trânsito de modo a mudarem para

verde quando um autocarro passa para chegar mais rapidamente às paragens de autocarro [34].

Além de aplicações, podemos falar da computação ubíqua [42], esta está a ser adaptada pelos fabricantes de automóveis na sua conceção do futuro automóvel com o objetivo de proporcionar às pessoas com deficiência visual, uma melhor segurança na condução.

2.3.3 Casas inteligentes

O conceito de casa inteligente pode ser definido como uma casa automatizada avançada ou controlada, uma vez que utiliza tecnologias de Inteligência Artifical para se tornar dinâmica, mais inteligente e aprender com as atividades diárias dos utilizadores, permitindo assim que os deficientes visuais vivam independentemente [43].

Na área de saúde, as casas inteligentes transformarão a forma como os serviços de saúde são prestados a todos os residentes. Pessoas com deficiência visual poderão ligar-se remotamente à sua clínica preferida dentro da sua casa, o que por vezes, acaba por evitar deslocações desnecessárias.

Algumas das formas como as casas inteligentes ajudam com a mobilidade dentro da casa da pessoa invisual é através do uso de aparelhos construídos para o efeito. Estes aparelhos podem ser utilizados através de comandos de voz que permitem à pessoa controlar vários aspetos e outros aparelhos de casa sem muito esforço.

Alguns dos aparelhos mais populares para este efeito são a *Amazon Echo Dot* ou a *Google Home* [30]. Através destes aparelhos o utilizador pode receber também notificações sobre vários eventos que se vão passando dentro de casa sem a necessidade de se deslocar ao local, como, por exemplo, fugas de água ou gás, ter deixado o fogão ligado ou a porta do frigorífico aberta e toques à campainha. Para além disso, e se tal for possível, podem-se tomar as medidas adequadas à notificação recebida sem se deslocar ao local.

Além disso, estas tecnologias também podem facilitar a vida de uma pessoa com deficiência visual ligando e desligando as luzes através de sensores infravermelhos, controlando as janelas e se estão abertas ou não através de motores e sensores adequados e controlar eletrodomésticos tais como frigoríficos, fornos e microondas.

Capítulo 3

Enquadramento conceptual

Este capítulo apresenta os vários tópicos e tecnologias relevantes ao desenvolvimento deste projeto, explicando conceitos base de funcionamento das tecnologias utilizadas no decorrer deste desenvolvimento, assim como a explicação de processos relevantes para o desenvolvimento da solução que irá ser apresentada.

É elaborada uma breve introdução ao sistema operativo *Android*, usado para o desenvolvimento da aplicação. Também são referenciadas duas linguagens de programação - o *Java* e o *Kotlin* - ambas possíveis de usar atualmente no desenvolvimento de aplicações android, e o *Android Studio*, o ambiente de desenvolvimento utilizado para a conceção do protótipo.

A principal forma de funcionamento da aplicação irá basear-se na geolocalização do utilizador através do reconhecimento de pontos de referência geográficos. Como tal, vai ser feita uma exposição do que são, incluindo uma explicação do que se considerou um ponto de referência e dos vários tipos que existem.

Finalmente, será apresentada a tecnologia de reconhecimento de imagem. Aqui são apresentadas redes neuronais, o conceito base desta tecnologia, e o *OCR*, responsável, especificamente, pelo reconhecimento de caracteres.

3.1 Sistema operativo Android

O *Android* é um sistema operativo baseado no Kernel e é atualmente desenvolvido pela *Google*. Apesar de o Android ter sido inicialmente lançado para telemóveis, hoje em

dia a sua aplicação excede este tipo de dispositivo. Atualmente o sistema é aplicado em tablets, leitores MP4 e ainda televisões com serviço de internet [6].

O sistema operativo é constituído por quatro camadas: aplicações, framework para aplicações, bibliotecas e *Linux Kernel* como mostradas na figura 3.1.

A camada de **Aplicações** é a camada superior da arquitetura *Android* onde estão localizadas todas as aplicações que são executadas sobre o sistema operativo, tais como mapas, calendários, contactos, entre outros.

A camada **Framework para aplicações** é a camada que permite que os programadores acedam às diversas funcionalidades do sistema para a criação de aplicações para *Android*. Esta inclui classes e serviços necessários para o desenvolvimento de aplicações Android. Existem uma série de gestores, como o gestor de atividades que efetua a gestão e o controlo de todas as atividades, gestor de notificações que permite que todas as aplicações mostrem alertas na barra de estados, gestor de localização que permite também a ativação de alertas quando um utilizador entra ou sai de uma localização em particular. Temos o Fornecedor de conteúdos que permite que uma aplicação tenha acesso a outros dados de aplicações, ou seja, haja partilha de dados.

Na camada **Bibliotecas** existem bibliotecas escritas em C/C++ utilizadas em vários componentes do sistema *Android*. A partir da camada *Framework para aplicações* pode-se ter acesso a estas bibliotecas. Existe a biblioteca SQLite para trabalhar com base de dados, biblioteca SSL responsável pela segurança da Internet, bibliotecas para reproduzir e gravar áudio ou vídeo.

Na camada **Android Runtime** é instanciada a máquina virtual Dalvik, que permite compilar as aplicações. Outro dos componentes do *Android Runtime* são as bibliotecas centrais, que disponibilizam uma API Java utilizada para programação.

Por fim existe a camada ***Linux Kernel*** que é o núcleo do sistema operativo Android. Funciona como abstração entre o hardware e as aplicações e é responsável pelos serviços principais do sistema operacional *Android*. Esta camada contém todos os drivers de hardware essenciais, como a câmara, teclado, entre outros.

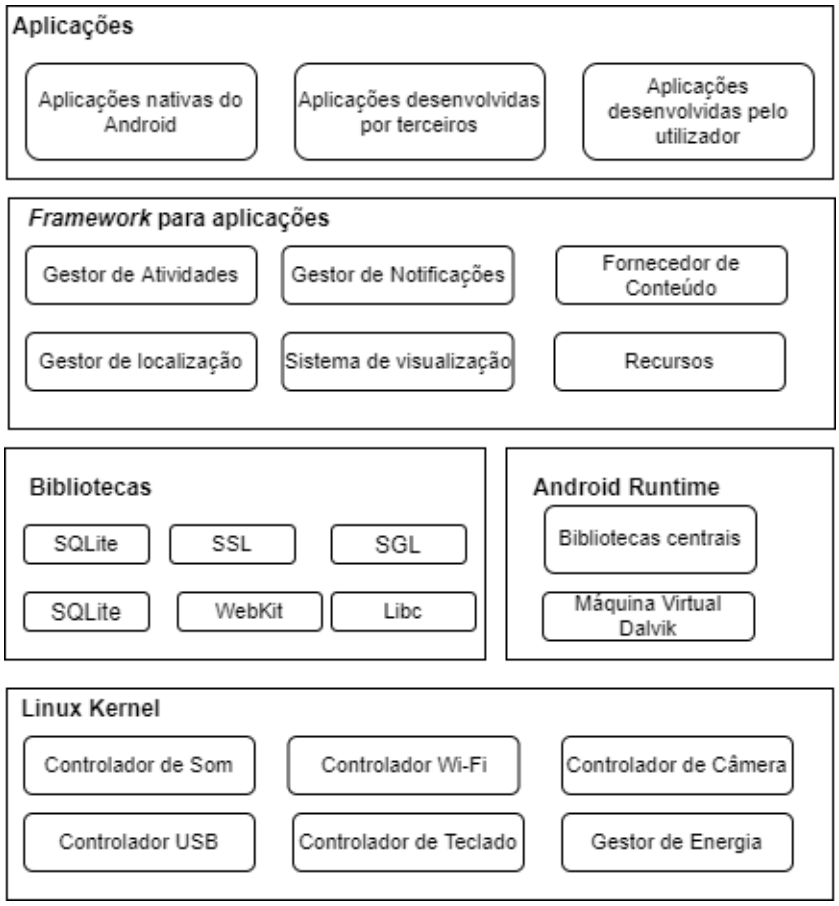

Figura 3.1: Arquitetura do Sistema Operativo Android (adaptado de [7]))

3.1.1 Acessibilidade

Uma das grandes limitações para as pessoas cegas e amblíopes é a dificuldade em aceder a informação escrita, obter orientação e apoio à mobilidade ou até mesmo ver emissões de televisão, filmes ou espectáculos.

Para ultrapassar estas limitações, o sistema *Talkback* [29] está embebido no sistema operativo da *Google*, tendo sido criado com o objetivo de ajudar as pessoas com deficiências visuais.

O *Talkback* é uma aplicação de acessibilidade para a leitura do ecrã dos telemóveis *Android*, facilitando a usabilidade de pessoas com deficiência visual e cegos. O retorno em voz é dado de acordo com o que aparece no ecrã do telemóvel e com o movimento dos dedos sobre a mesma. Esta já vem instalada de raíz. O *TalkBack* ajuda os cegos a ouvir o que estão a fazer com o seu telemóvel, uma vez que a aplicação diz-lhes o item que acabaram de selecionar ou escolher. A aplicação também pode ler textos em voz alta e cada movimento que o utilizador faz no seu telemóvel está a ser cuidadosamente monitorizado e falado pela aplicação.

Foi criada uma nova versão em Fevereiro de 2021 [60] de modo a que facilite ainda mais o processo a este segmento de pessoas.

Com esta nova versão, é possível realizar uma série de novos gestos que detetam movimentos, havendo também a possibilidade de selecionar um texto específico. Outra vantagem é de o utilizador poder pedir para que o *Talkback* leia um texto mais rápido ou devagar.

3.1.2 Java e Kotlin

Para o desenvolvimento de uma aplicação *Android* existem atualmente duas linguagens possíveis: Java [22] e Kotlin [9], sendo que Kotlin é a oficial desde o ano de 2017.

A linguagem *Java* tem duas particularidades muito relevantes: Orientada a Objetos e Estaticamente Tipada. Uma linguagem orientada a objetos é baseada na modelagem de objetos e na comunicação entre eles. Uma linguagem Estaticamente Tipada significa que é necessário declarar todas as variáveis, isto é, todas estas tenham um tipo de dados.

Por outro lado, existe a linguagem *Kotlin* que atualmentemente é a oficial utilizada

para o desenvolvimento *Android*. Algumas das vantagens apresentadas são o facto da linguagem ser mais legível, o código ser mais seguro contra os valores nulos nas varíaveis e ter a possibilidade de declarar as variáveis como mutáveis ou imutáveis. Por outro lado, uma das desvantagens encontradas é que o tamanho final dos projetos é maior quando comparado com um projeto desenvolvido em Java. Isto ocorre porque o *Kotlin* tem a sua própria biblioteca que é adicionada à aplicação.

3.1.3 Android Studio

O *Android Studio* é o IDE (Ambiente de desenvolvimento integrado) utilizado para o desenvolvimento de projetos Android [8].

O Android Studio oferece um par de funcionalidades para aumentar a produtividade e criar aplicações Android facilmente, como por exemplo:

- Um sistema de construção flexível baseado no Gradle.
- Um ambiente unificado para poder desenvolver para todos os dispositivos Android.
- Execução instantânea para aplicar alterações a aplicações em execução sem a necessidade de compilar um novo APK.
- Modelos de código e integração com GitHub para ajudar a construir características comuns de aplicações e importar amostras de código.
- Ferramentas e estruturas de teste
- *Lint*, uma ferramenta suspeita de verificação de código para detetar desempenho, usabilidade, compatibilidade de versões
- IDE baseado em IntelliJ IDEA.

A interface para o desenvolvimento do Android Studio permite visualizar o design da tela que está a ser criada. Isso permite ao programador implementar o código e verificar o resultado da interface do projeto, o que evita a execução do projeto a cada verificação de tela, diminuindo assim o tempo de produção da aplicação .

3.2 Pontos de referência

O termo ponto de referência pode significar muitas coisas para pessoas diferentes, mas para orientação e mobilidade, são necessárias algumas características definidas para considerar um objeto como um ponto de referência. Pode ser um objeto fixado, isto é, o objeto tem de ser inamovível e fixado num local fixo. Alguns bons exemplos incluem postes telefónicos, edifícios, ou árvores. Alguns maus exemplos incluem um carro estacionado, latas de lixo móveis, ou móveis não fixos no pátio. Pode também ser reconhecido como um objeto único, isto é, o objeto precisa de ser algo facilmente identificável visualmente, audivelmente, ou taticamente e também único a uma área. Bons exemplos incluem uma única boca-de-incêndio ao longo de uma rua, uma placa de paragem numa esquina, ou um corrimão ao longo de um caminho que conduz a uma casa. Os maus exemplos incluem: uma de muitas árvores semelhantes alinhadas ao longo de uma calçada, uma porta para uma sala de aula universitária que é a mesma que as outras portas da sala de aula [69].

3.3 Reconhecimento de imagem

O reconhecimento de imagem torna-se importante para este projeto, dado ser uma necessidade para as pessoas cegas e amblíopes. Atualmente é uma prática usada em vários âmbitos, tais como: reconhecimento de face, reconhecimento de código de barras e QR-Code, entre outras.

Alguns dos seus usos mais recentes no dia a dia focam-se na área da saúde, como na deteção de doenças como tumores, AVCs através da análise de várias imagens médicas, melhorando assim as tecnologias de diagnóstico destas doenças. É usado na área de fabrico, no controlo da qualidade na fabricação de produtos de modo a reduzir os defeitos. Também se pode encontrar na área da condução, na deteção de obstáculos ou objetos, o que ajuda na previsão de velocidades e no comportamento de outras entidades.

Numa primeira fase irão ser explicados resumidamente modelos de redes neurais profundas e alguma das suas características.

Noutra fase irá haver um foco maior sobre o OCR devido ao facto de ser uma tecnologia essencial para as pessoas cegas e amblíopes.

O uso desta tecnologia dá uma maior autonomia e independência à pessoa cega e

amblíope.

3.4 Modelos de Redes Neurais

A maioria das melhorias na deteção de objetos está associada a novas representações de objetos e modelos de *Deep Learning*.

Uma rede neural profunda é uma rede neural artificial com várias camadas ocultas entre as camadas de entrada e saída. Semelhante às redes neurais artificiais rasas, as redes neurais profundas podem modelar relacionamentos não lineares complexos.

As redes neurais são amplamente usadas na aprendizagem supervisionada e nos problemas de aprendizagem por reforço. Essas redes são baseadas num conjunto de camadas conectadas entre si.

O principal objetivo de uma rede neural é receber um conjunto de entradas, executar cálculos progressivamente complexos sobre elas e fornecer saída para resolver problemas do mundo real, como classificação.

As Redes Neurais apresentam grandes diferenças em relação aos métodos tradicionais, uma vez que têm a capacidade de aprender modelos complexos e representações de objetos robustos sem a necessidade de modelos e características desenhados à mão.

3.4.1 Rede Neural Convolucional

No contexto de inteligência artificial e *Deep Learning*, as redes neurais convolucionais, também conhecidas como ConvNets, baseiam-se no córtex visual, voltada para o desenvolvimento do campo de visão computacional, para classificação de imagens.

Estas pertencem a uma categoria de algoritmos baseados em redes neurais artificiais que utilizam a convolução em pelo menos uma das suas camadas. O processo de classificação consiste em dada uma imagem de entrada, a saída seja a classe a qual a imagem de entrada pertence ou a probabilidade de pertencer a determinada classe. O pré-processamento exigido num ConvNet é muito inferior em comparação com outros algoritmos de classificação. Enquanto nos métodos primitivos os filtros são concebidos à mão, com formação suficiente, as ConvNets tem a capacidade de aprender estes filtros.

3.4.2 Redes Neurais Convolucionais Regionais

A Rede Neural convolucional Regional (R-CNN) foi proposta por investigadores da AI na Universidade da Califórnia, Berkley, em 2014 [26]. Esta é composta por três componentes-chave.

Primeiro, um selector de regiões usa "pesquisa selectiva", um algoritmo que encontra regiões de pixels na imagem que podem representar objetos, também chamadas "regiões de interesse"(RoI). O seletor de região gera cerca de 2.000 regiões de interesse para cada imagem.

Em seguida, "as regiões de interesse"são deformadas num tamanho pré-definido e passadas para uma rede neural convolucional. A rede neural convolucional processa cada região separadamente extrai as características através de uma série de operações de convolução. Esta utiliza camadas totalmente ligadas para codificar os mapas de características num vetor unidimensional de valores numéricos.

Finalmente, um modelo de aprendizagem da máquina classificadora mapeia as características codificadas obtidas da rede neural convolucional para as classes de saída. O classificador tem uma classe de saída separada para "fundo", que corresponde a qualquer coisa que não seja um objeto.

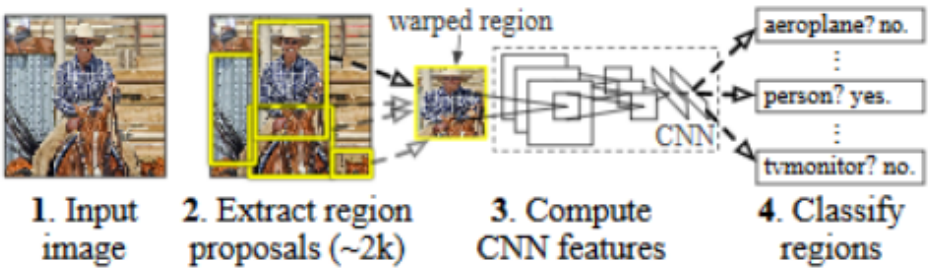

Figura 3.2: Deteção de objetos via RCNN (Fonte: [26])

3.4.3 Fast R-CNN

Em 2015, o autor principal do artigo *Region Based Convolutional Neural Networks* (R-CNN) propôs uma nova arquitectura chamada Fast R-CNN [25], que resolveu alguns dos problemas do seu predecessor. *Fast R-CNN* traz a extração de características e a

seleção da região num único modelo de aprendizagem da máquina.

Fast R-CNN recebe uma imagem e um conjunto de regiões de interesse e devolve uma lista de caixas de delimitação e classes dos objetos detectados na imagem.

Uma das principais inovações no Fast R-CNN foi a "camada de agrupamento de RoI", uma operação que leva os mapas de características da rede neural convolucional e as regiões de interesse para uma imagem e fornece as características correspondentes para cada região. Isto permitiu ao Fast R-CNN extrair características para todas as regiões de interesse na imagem numa única passagem, em oposição ao R-CNN, que processou cada região separadamente. O resultado foi um aumento significativo da velocidade.

No entanto, uma questão permaneceu por resolver. O *R-CNN* rápido ainda exigia que as regiões da imagem fossem extraídas e fornecidas como entrada para o modelo. O Fast R-CNN ainda não estava pronto para a deteção de objetos em tempo real.

3.5 OCR

O OCR é uma tecnologia que permite reconhecer caracteres. Permitem converter tipos diferentes de documentos digitalizados em dados pesquisáveis ou editáveis, ou seja, convertem imagens de texto em texto real.

Há dois métodos para realizar OCR: correspondência matricial e deteção de características. A correspondência matricial é a mais simples das duas; toma uma imagem e compara-a com uma biblioteca existente de matrizes de caracteres ou modelos para gerar uma correspondência. A deteção de características é mais complexa, pois procura características gerais como linhas diagonais, curvaturas, intersecções e compara-a com outras características da imagem dentro de uma certa distância [50].

Para fazer uma leitura otimizada de documentos, são realizadas três etapas [36].

Estas são:

- Pré-Processamento
- Reconhecimento
- Tratamento

A etapa de Pré-Processamento consiste em preparar a imagem para o reconhecimento de caracteres e para isso são eliminadas ou detetadas todas as características das imagens que não são caracteres, como ícones ou marcas de água. São convertidas em imagem binária.

O reconhecimento de caracteres pode ser feita de diversas formas, uma delas consiste na comparação de cada carácter identificado previamente com uma base de símbolos para definir padrões e encontrar semelhanças. A outra técnica consiste em procurar características gerais como volume, linhas e curvas.

Por fim, depois da identificação e definição de caracteres, a tecnologia OCR compara esses dados com uma base de palavras do idioma, ou com o padrão sequencial dos números de documentos.

Em 2021, onde tudo está a tornar-se digitalizado e avançado, a tecnologia OCR está a ser utilizada por várias empresas para simplificar os processos empresariais, melhorar a acessibilidade, e aumentar a satisfação do cliente. Abaixo, alguns dos casos de utilização mais destacados de OCR hoje em dia.

O reconhecimento automático de matrículas utiliza a tecnologia OCR para identificar os números nas matrículas. Atualmente, o reconhecimento de matrículas é utilizado num conjunto diversificado de aplicações comerciais para encontrar carros roubados, calcular taxas de estacionamento, faturar portagens ou para controlo de acesso a zonas de segurança.

A indústria bancária é considerada um dos maiores consumidores de tecnologia OCR,

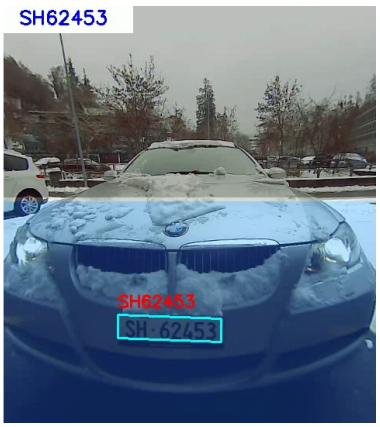

Figura 3.3: Exemplo de reconhecimento automático de matrículas em aplicações de gestão de parques de estacionamento (Retirado de [49])

visto que ajuda a aumentar a segurança, melhora a gestão de dados, otimiza a gestão de risco e melhora a experiência do cliente.

Antes de aplicar a tecnologia OCR, a maioria dos documentos bancários eram físicos, incluindo registos de clientes, cheques, extratos bancários, e outros. Com a tecnologia de OCR, tornou-se possível digitalizar e armazenar até documentos mais antigos em bases de dados.

Um exemplo disto pode ser visto em aplicações bancárias móveis, onde os cheques podem ser depositados digitalmente e processados dentro de dias através de funções de depósito de cheques baseadas em OCR. Há também a possibilidade de aumentar a segurança, pelo facto do depósito eletrónico de cheques através da tecnologia OCR resultar na prevenção da fraude e em transações cada vez mais seguras, promovendo uma melhor experiência do utilizador.

Na área de saúde, a tecnologia OCR também pode ser usado, pois, permite o acesso digital aos historiais médicos dos pacientes, tanto pelos pacientes como pelos médicos.

Além disso, os registos de pacientes, incluindo os seus raios X, tratamentos, testes, registos hospitalares e pagamentos de seguros, podem ser facilmente digitalizados, pesquisados e armazenados utilizando esta tecnologia.

Assim, o OCR ajuda a racionalizar o fluxo de trabalho e reduzir o trabalho manual nos hospitais, mantendo os registos atualizados.

Capítulo 4

Revisão da literatura

Este capítulo apresenta a revisão da literatura referente a várias aplicações que são desenvolvidas para pessoas com deficiências visuais, tanto em ambiente interior como em ambiente exterior. São apresentadas algumas dessas propostas, com o objetivo principal de compreender as metodologias, tecnologias e abordagens. No final será apresentado uma tabela comparativa sobre as mesmas.

4.1 Navegação de pessoas cegas e amblíopes

Em [41], é proposto um sistema exterior no qual é utilizada uma base de dados geográfica de percursos pedonais numa cidade e também um GPS para fornecer a posição do utilizador, no qual o objetivo é determinar o percurso óptimo desde a origem até ao destino.

Em [64], os autores apresentam um sistema de navegação móvel para ajudar na navegação de pessoas com deficiências visuais ao navegar em ambientes exteriores. A solução sugerida baseia-se na utilização de uma câmara de dispositivo móvel e algoritmos *Deep Learning* para reconhecer e detetar diferentes objetos, bem como fornecer informações adicionais para ajudar.

Em [15], os autores apresentam um aplicação de nome "BlindNavi"que pretende facilitar a vida às pessoas com deficiências visuais. O principal objetivo é fornecer uma nova solução de ajuda à mobilidade, sob a forma de uma aplicação de navegação que armazena informações significativas durante a viagem para torná-la mais segura. A aplicação

utiliza feedback de voz que consiste em tacos multissensoriais combinados com tecnologia de microlocalização para ajudar pessoas com deficiências invisuais a explorar o ambiente exterior.

Em [14], os autores sugerem uma aplicação de navegação que recorda informações significativas durante a viagem e torna a viagem mais segura. Usam um feedback de voz que consiste em pistas multissensoriais combinadas com a tecnologia de microlocalização para ajudar os deficientes visuais a sair das suas casas e explorar o ambiente exterior em segurança.

Em [4], a aplicação fornece informação em tempo real sobre onde se encontra o utilizador, qual a direção que este está a tomar, e outras informações sobre o meio envolvente. A aplicação utiliza sinais *bluetooth*.

Em [73], os autores apresentam um sistema de navegação interior para pessoas com deficiências visuais, utilizando uma abordagem baseada na visão por computador. Marcadores fiduciais com informação em forma de aúdio são colocados no ambiente interior. Este sistema de navegação fornece dois tipos de orientação: orientação de modo livre e navegação de modo orientada. Este primeiro, o utilizador navega livremente e o sistema apenas informa-o sobre a sua posição atual através de aúdio com base no marcador fiducial. No modo de navegação orientada, o sistema utilizada o algoritmo *Dijkstra*, em que o utilizador é guiado desde o início até ao seu destino pelo caminho mais curto.

Em [24], os autores descrevem um sistema de navegação interior utilizando uma bengala modificada que inclui sensores de cor e um leitor Identificação por radiofrequência (RFID), juntamente com um microcontrolador, um altifalante e um equipamento de vibração. Os sistemas instalados sobre a bengala são um sistema de navegação e um sistema de informação cartográfica. O sistema de navegação segue uma linha de navegação colorida que é colocada no chão. Um sensor de cor é instalado na ponta de bengala e deteta uma cor da linha que o utilizador está a percorrer sendo que o mesmo é informado que está a percorrer sobre a linha através da sensação de vibração.

Em [12], os autores apresentam um sistema de navegação que deteta os obstáculos e ajuda na navegação de pessoas com deficiência visual sobre o melhor caminho a seguir. O obstáculo é detetado através de um sistema de deteção baseado em infravermelhos e este envia o feedback para o recetor através de um feedback vibro-táctil ou sonoro para

informar o utilizador sobre a sua posição. Um sensor é fixo sobre a cabeça do utilizador e permite que o utilizador obtenha informação sobre os obstáculos. Uma das grandes limitações é na falta de reconhecimento de objetos.

Em [63], os autores apresentam um sistema que integra os dados de um Sistema de informação geográfico (SIG) de um edifício com deteção de pontos de referência para localizar o utilizador no edifício e para traçar e validar uma rota para a navegação do utilizador. Assim, o sistema desenvolvido complementa a cana branca para melhorar a autonomia do utilizador durante a navegação interior.

Em [35], os autores propõem uma aplicação *Android* para ser usada na navegação em ambiente interior com instruções em forma de áudio onde são utilizadores códigos QR. Esta fornece assistência na navegação em caminhos pré-definidos para cegos. Estes códigos QR são colocados em secções de pisos diferentes que após uma distância específica atua como entrada para a deteção e a navegação da localização atual. Sempre que esse código QR é digitalizado, fornece ao utilizador a informação da localização atual e pede ao utilizador para selecionar o destino, em que disponibiliza depois o caminho mais curto e ótimo utilizando algoritmos de localização de caminho. Caso seja detetado um caminho diferente, o utilizador é alertado e guiado de volta ao caminho correto. Todas as instruções ao longo da aplicação são fornecidas em forma áudio ao utilizador.

Em [37] , os autores propõem um sistema de navegação utilizando uma aplicação móvel que lê alvos coloridos obtidos através da câmara integrada. A aplicação Android desenvolvida chamada "GuiderMoi"é utilizada principalmente para ajudar os cegos em ambientes e edifícios interiores. Na etapa de deteção de cor do alvo, usam algoritmos de reconhecimento de cor principalmente Camshift [38].

Em [40], os autores introduzem um sistema de navegação para as pessoas com dificuldades visuais para ambientes exteriores e interiores. Os componentes deste sistema incluem um laser que fornece a distância e o ângulo obtidos de diferentes posições, um computador portátil, auscultadores e uma unidade de medição inercial que normalmente contém acelerómetros, giroscópios e magnetómetros que podem estar localizados em qualquer parte do corpo da pessoa com deficiência visual. Os algoritmos usados neste estudo são o Localização e Mapeamento Simultâneos (SLAM), o Pedestrian Dead Reckoning (PDR) e o GPS.

Em [72], os autores descrevem um sistema de navegação integrado com RFID e GPS, Smart-Robot para os deficientes visuais. O Smart Robot utiliza a localização baseada em RFID e GPS funcionando tanto em ambiente interiores como em ambientes exteriores. Este orienta o utilizador para um destino pré-definido ou criar uma rota em tempo real para utilização posterior. Em modo de navegação, o Smart Robot chega ao destino evitando obstáculos utilizando sensores ultrassónicos e infravermelhos.

Em [65], os autores propõem um sistema para os deficientes visuais tanto em ambiente interior como exterior que se concentra principalmente em dar uma saída baseada na voz para a prevenção de obstáculos e também para a navegação utilizando um sensor ultrassónico. Também utilizam o GPS e voz que alertam os deficientes visuais.

Em [68], os autores apresentam um sistema de navegação baseado na visão para orientar as pessoas com deficiências visuais em ambientes exteriores e interiores. O sistema de posicionamento usa uma câmara montada que tira fotos do ambiente e um algoritmo para combinar em tempo real referências particulares extraídas da imagem, com pontos de referências 3D já armazenadas no sistema. Tem as suas limitações, visto que não é adequado para ser utilizado em ambientes que são usados pela primeira vez.

Em [66], foi apresentada um sistema de rastreio incorporado para ajudar a pessoa cega a navegar e também para fornecer a facilidade para seguir o movimento de uma pessoa cega e assegurar se está perdido. Foi implementado com uma aplicação Android utilizando um sistema GPS. O sistema proposto foi experimentado com taxa de erro muito mínima.

Nas tabelas 4.1, 4.2, 4.3 descrevem-se propostas para os desafios previstos em ambientes interiores ou exteriores.

Para soluções interiores, existem abordagens como Landmark, Dead Reckoning e SLAM. O *Dead Reckoning* é uma alternativa ao GPS. Este permite calcular o posicionamento com alta precisão através da utilização de informação de vários sensores (sensor giroscópico, odómetro, acelerómetro) para calcular a posição atual [48]. As vantagens de se usar este sistema incluem o baixo custo e a precisão em que é estimada a posição em tempo real[58].

Para soluções exteriores, temos aplicações focadas no uso de Deep Learning, *feedback de voz*, microlocalização e GPS.

De modo geral, estas soluções, tanto para ambientes interiores como para ambientes exteriores, têm como objetivo melhorar a navegação de utilizadores cegos e amblíopes.

Estudo	**Abordagem**
EXTERIOR	
[41]	SIG, GPS
[64]	Deep Learning, voz
[15]	Microlocalização e *feedback* de voz
[14]	Microlocalização e feedback de voz
[4]	Sinais bluetooth

Tabela 4.1: Aplicações exteriores

Estudo	**Abordagem**
INTERIOR	
[73]	Marcadores Fiduciais em forma de áudio
[24]	Sensores de cor, RFID
[12]	Infrared based detecting system
[63]	SIG, landmark
[35]	Códigos QR, Aúdio
[37]	Camshift

Tabela 4.2: Aplicações interiores

Estudo	**Abordagem**
INTERIOR/EXTERIOR	
[40]	SLAM, PDR e GPS
[72]	RFID, GPS, sensores infravermelhos
[65]	Sensor ultrassónico e voz

[68]	Pontos de referência 3D
[66]	GPS

Tabela 4.3: Aplicações interiores/exteriores

4.2 Posicionamento exterior usando pontos de referência

São vários os artigos que abordam a problemática da navegação usando pontos de referência.

Em [59] , os autores abordam um sistema que utiliza técnicas de correspondência de modelos para encontrar múltiplos padrões de referências numa imagem para estimar a distância até ao ponto de referência.

Em [11] , foi desenvolvido um sistema que utiliza QR Codes afixados já em pontos de referência registados. Isto permite que o utilizador se localize em relação à sua rota e com instruções de navegação dadas em termos desses pontos de referência. O sistema inclui imagens de cada ponto de referência, ajudando os utilizadores a navegar visualmente. Funciona como um dispositivo móvel, sendo que os utilizadores utilizam a câmara do seu dispositivo para registar a cada ponto de referência, o QR code e assim atualizar a sua posição.

Em [47], foi apresentado um método para a localização em ambientes exteriores utilizando pontos de referência baseados na aprendizagem. O método de localização proposto baseia-se em Faster R-CNN (Faster Regional Convolutional Neural Network) que é implementado para a deteção de pontos de referência em imagens e o FFNN (Feed forward neural network) que foi utilizado para recuperar as coordenadas de localização e orientações da bússola do dispositivo implementado no mundo real com base em pontos de referências detetados desde o R-CNN.

Em [44], os autores relatam como conduzir um robot móvel utilizando pontos de referência naturais como árvores e plantas no campus exterior da Universidade. Primeiramente propõe uma aquisição natural do ponto de referência pelo Landmark Agent (LmA).

Este Landmark Agent tem três estados: SLEEP, WAKE UP, TRACK. O estado SLEEP quer dizer que não há nenhum candidato a ponto de referência. WAKE UP significa que o candidato a ponto de referência seja detetado e o agente começa a localizá-lo e por fim, o estado TRACK significa que o objeto é repetidamente detetado duas ou mais vezes. Quando faz esta ordem de transição: SLEEP – WAKE UP – TRACK, o ponto de referência é detetado e gravado no mapa do LmA. Depois propõe uma navegação autónoma utilizando os pontos de referência naturais adquiridos. A estratégia de navegação que usam é a navegação utilizando o Perceived Route Map (PRM) gerados pela aquisição automática de ponto de referência naturais através do ensino das rotas.

Em [55] , o autor propõe uma abordagem para gerar instruções de rotas baseados em pontos de referências, utilizando um conjunto de funções de avaliação para avaliar o visual, a semântica e a saliência estrutural de objetos espaciais individuais, por exemplo, o edifício.

No próximo capítulo irá ser mostrado como foi desenvolvida a aplicação, passando pela sua arquitetura, assim como a escolha da framework.

Capítulo 5

Desenvolvimento da aplicação

Neste capítulo, numa primeira fase será apresentada a arquitetura onde será explicado como solucionar o problema identificado, com recurso ao uso do *Talkback*. Na fase seguinte, vai ser feita a comparação da escolha de framework de reconhecimento de imagem, mostrando particularidades de cinco frameworks. No fim irá ser apresentado uma tabela comparativa das mesmas.

5.1 Arquitetura

Esta secção apresenta a arquitetura conceptual proposta para a abordagem ao problema identificado (posicionamento de pessoas cegas e amblíopes com base em pontos de referência quando perdem a orientação) assim como um estudo sobre APIs de reconhecimento de imagem, para que fosse possível escolher a mais adequada para este sistema.

Na figura 5.1 ilustra-se a arquitetura conceptual definida para o sistema. O sistema funciona da seguinte forma: o utilizador, utilizando a aplicação móvel, deve apontar a câmara para um determinado local que o rodeia e, automaticamente, é tirada e enviada uma fotografia através da aplicação para a API de reconhecimento de imagem do Google Cloud Vision.

Após o processamento, o resultado é devolvido e a pessoa cega é informada do resultado que é transmitido via áudio através do TalkBack.

O resultado permitirá à pessoa identificar a localização onde se encontra, o que representa informação muito relevante para que uma pessoa cega e amblíope seja capaz de se

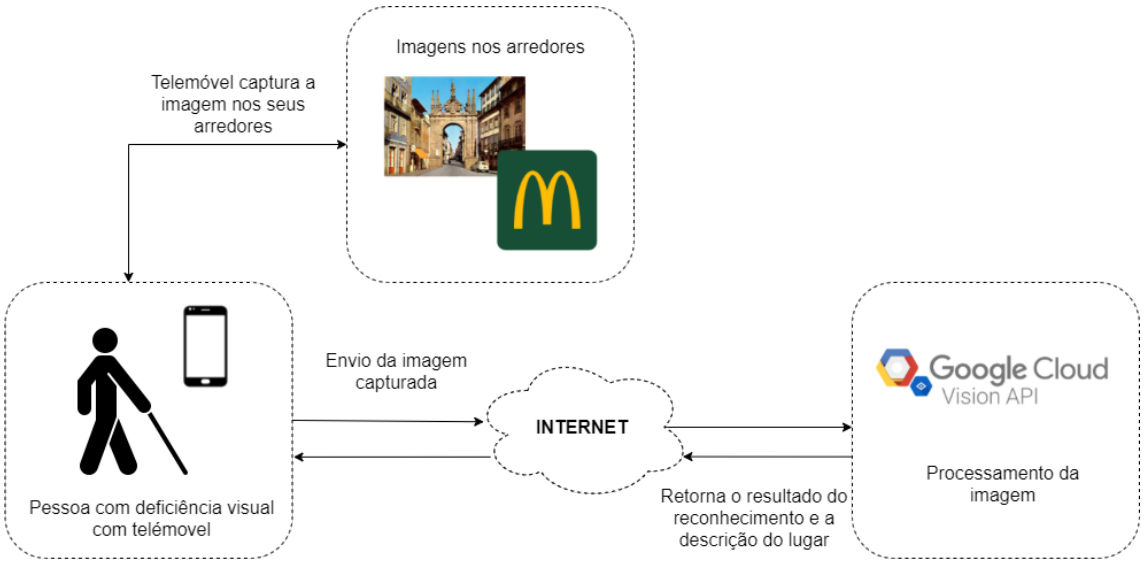

Figura 5.1: Diagrama de arquitetura

orientar caso tenha perdido a orientação.

O algoritmo de reconhecimento de imagem está atualmente preparado para processar logótipos, pontos de referência ou texto (em que um exemplo são o nome de lojas).

5.2 Escolha da framework de reconhecimento de imagem

Hoje em dia, o reconhecimento de imagens é uma prática utilizada em vários âmbitos, tais como: reconhecimento facial, reconhecimento de código de barras e QR-Code, análise de sentimentos, entre outras.

De modo a tomar a decisão entre as frameworks escolhidas, é apresentada no final desta secção, uma tabela com as funcionalidades de cada API.

5.2.1 Google Cloud Vision API

O Google Cloud Vision API [17] é uma ferramenta que a Google fornece no modo Beta aos criadores de aplicações ou empresas que vêem potencial no serviço de reconhecimento de imagem, para identificar os diferentes objetos de uma imagem.

Esta ferramenta classifica as imagens em categorias, e é capaz de detetar objetos e rostos em fotografias, além de ser capaz de ler o texto encontrado nas imagens utilizando a tecnologia OCR.

A API permite classificar as imagens a partir dos seguintes recursos:

- **LABEL_DETECTION** - Pode detetar e extrair informação sobre entidades numa imagem, através de um amplo grupo de categorias. Poderá fornecer rótulos relevantes, como, por exemplo, categorias, palavras-chave, objetos gerais.
- **TEXT_DETECTION** - Realiza OCR. Com isto podemos extrair texto impresso ou manuscrito a partir de imagens, como fotos de sinais de rua, faturas, nomes de ruas.
- **FACE_DETECTION** - A deteção facial deteta vários rostos dentro de uma imagem com os principais atributos faciais associados, como estado emocional.
- **LANDMARK_DETECTION** - A deteção de pontos de referência deteta estruturas naturais e artificiais numa imagem. Detecta marcos geográficos conhecidos..
- **LOGO_DETECTION** - Deteta logótipos de marcas famosas, como, por exemplo, Google, Burguer King.

Esta API permite realizar uma única chamada que retorna os recursos todos de uma vez, com um único upload, embora o resultado seja obtido mais rapidamente quando os recursos são executados individualmente.

Não tem nenhum custo inicial, sendo um serviço pago à medida que se vai gastando os recursos disponíveis, sem taxas de recisão. O tempo de resposta da API é muito rápido, o que é um fator importante.

5.2.2 IBM Watson Visual Recognition API

O Visual Recognition API [33] utiliza algoritmos de aprendizagem profunda para entender os conteúdos de imagens, análise de imagens para cenas, objetos, cores, comida e outros assuntos que podem fornecer contributos para o seu conteúdo visual.

Esta API apresenta algumas vantagens, como tratar dados em grandes quantidades, o processamento de dados não estruturados e pode ser utilizado como um sistema de apoio à decisão.

Por outro lado, o custo de manutenção é elevado e embora possa haver o processamento de dados não estruturados, este não é feito de forma direta. A API também acabou por ser descontinuada em Janeiro de 2021 [32].

5.2.3 Clarifai API

A Clarifai [16] utiliza Rede neural convolucional (CNN), um subcampo de aprendizagem de máquinas, para compreender objetos em imagens e vídeos. Utiliza uma extensa biblioteca de terminologias semânticas e visuais para inteligência artificial. Também usa semelhanças semânticas e visuais para comparar imagens carregadas com outras imagens na biblioteca para mostrar as semelhanças. Utiliza Inteligência Artificial com visão computacional, o que proporciona eficiência nos processos empresariais.

É também capaz de detetar conteúdo explícito, identificar celebridades, e reconhecer rostos. O Clarifai pode também determinar a cor dominante de uma imagem.

Existem dois modelos para a incorporação de rostos ou artigos gerais. Baseiam-se em modelos de deteção facial e modelos gerais, respetivamente. Existe também um modelo para verificar se a imagem contém conteúdos inseguros como drogas ou nudez.

Apesar de ser uma API de fácil integração, acaba por ser demasiado expensivo, lento e poderemos ter de entrar possíveis problemas de privacidade.

5.2.4 Microsoft Computer Vision API

Microsoft Computer Vision API [10] está alojado no Microsoft Azure e fornece aos programadores, acesso a algoritmos avançados de processamento de imagem, devolvendo informação após análise das imagens. Ao carregar uma imagem ou especificar um URL de imagem, os algoritmos de Microsoft Computer Vision podem analisar o conteúdo visual de diferentes maneiras com base em entradas e escolhas do utilizador.

A Computer Vision API permite classificar o conteúdo da imagem fornecendo uma lista abrangente de etiquetas e tentando construir uma descrição da cena em linguagem natural. Além disso, o API é capaz de reconhecer as celebridades e os pontos de referência.

Outra característica é o OCR de texto impresso e como uma pré-visualização. O OCR para os textos manuscritos também está disponível, mas no entanto apenas para a língua inglesa.

Esta framework está integrada com Microsoft Azure, base de dados SQL, pelo que pode ser empacotado como uma solução. O tempo de resposta também é muito rápido.

Por outro lado, se ultrapassar o número de transações acima do esperado, não vai dar

a melhor resposta possível e exige assinatura Azure, ou seja, não é gratuito.

5.2.5 Amazon Rekognition

A Amazon Rekognition [56] é um dos principais fornecedores de serviços para adicionar uma análise visual poderosa às suas aplicações. É fornecido como uma API tanto para imagens como para vídeos.

O reconhecimento pode compreender que objetos e pessoas estão no cenário e o que está a acontecer. Pode funcionar como um filtro de conteúdo para conteúdo adulto. Além disso, pode compreender o texto na imagem. Um dos poderes da Amazon Rekognition é a capacidade de detetar, reconhecer e identificar pessoas.

É capaz de identificar com precisão uma pessoa numa fotografia e num vídeo, utilizando um conjunto de dados privado de imagens faciais e também pode reconhecer pessoas famosas nas suas imagens.

É também capaz de analisar o sentimento, a idade, a presença de olhos e de touca, o cabelo facial e outras características. Para os vídeos, é possível acompanhar a mudança destas características ao longo do tempo.

Esta framework tem como vantagem a capacidade de analisar 5000 imagens por mês.

A API também apresenta algum atraso no tempo de resposta.

5.2.6 Comparação das APIs

A tabela 5.1 compara as opções do mercado atualmente existentes de forma a apurar as funcionalidades que mais se adequam aos requisitos identificados para este protótipo.

Para a nossa solução é necessário que a API nos permita o seguinte:

- Fazer upload de uma imagem, visto que será necessário obter uma captura de imagem através do telemóvel e fazer o reconhecimento da mesma através da API.
- Permitir o reconhecimento de texto é essencial, já que podemos detetar e extrair textos de um ponto de referência que não é facilmente detectável pela API. O suporte multi-linguístico é importante para garantir que a aplicação se adapta a outros países e outros idiomas.

API	Funcionalidades
Google Cloud Vision API	Reconhecimento de objetos Deteção de conteúdo explícito **Deteção de pontos de referência** **Deteção de logótipos** Devolver descrições das imagens Identificação da entidade Correspondência de imagens **Reconhecimento de texto**
Amazon Rekognition	Reconhecimento de objetos Deteção de conteúdo explícito Reconhecimento de Celebridades Captura de movimento **Reconhecimento de texto**
IBM Watson Visual Recognition	Compatível com a aprendizagem mecânica Vários modelos de aprendizagem de máquinas de identificação de objetos pré-carregados
Clarifai	Marcação automatizada de imagens Deteção facial Deteção de Celebridades Análise Demográfica **Deteção de logótipo**
Microsoft Computer Vision API	Deteção facial **Deteção de pontos de referência** Deteção de Celebridades **Reconhecimento de texto** Extracção de informação de documentos Propriedades da imagem Descrição e Categorização do Conteúdo da Imagem

Tabela 5.1: Funcionalidade das APIs

- Permitir o reconhecimento de logótipos, visto que é complementar ao reconhecimento de texto ou ponto de referência.
- Comparar imagens com pontos de referência conhecidos e que estejam previamente disponíveis através da API.

Dos critérios mencionados na tabela 5.1, a negrito encontram-se os critérios que assumem maior relevância para o projeto e tendo em conta as particularidades anteriormente referidas, são:

- Deteção de pontos de referência é importante, porque encontra estruturas famosas, naturais e construídas pelo homem numa imagem.
- Reconhecimento de texto - Esta funcionalidade é necessária, pois nós vamos passar por várias zonas que têm farmácias, CTT, bancos que não são reconhecidos automaticamente pela API. O reconhecimento textual, como alternativa secundária, auxiliará na deteção de localização do utilizador quando não é possível por imagem.
- Deteção de logótipos para identificar estabelecimentos com símbolos de empresas e marcas famosas.

Com base nestes critérios e visto que apenas uma das API cumpre os critérios todos mencionados, a ferramenta que se conclui ser a mais adequada é o Google Cloud Vision.

Após a escolha da API, o próximo capítulo irá evidenciar como foi desenvolvida a aplicação.

5.3 Desenvolvimento da aplicação móvel

5.3.1 Visão geral

Neste capítulo vão ser evidenciados os principais pontos do desenvolvimento da aplicação. O principal objetivo desta aplicação móvel é ajudar as pessoas cegas e amblíopes no posicionamento numa cidade, quando se encontram perdidas. Para isto, é necessário que a aplicação responda aos seguintes requisitos:

- Suportar o reconhecimento de texto;
- Suportar o reconhecimento de pontos de referência conhecidos;
- Suportar o reconhecimento de logótipos;

Numa primeira fase foi feita a integração da aplicação com a *Google Cloud Vision*, onde foram criados uma conta, um projeto e a chave da API onde se adicionam as dependências mostrada nas figuras 5.4 e 5.5.

Na fase seguinte foi planeado o *layout* da nossa aplicação com o objetivo de ter uma *Surface View*. Após termos o *layout* definido, passou-se para a integração da câmara na aplicação. Numa fase final foi realizada a integracão com a *Google Cloud Vision API*.

Como já foi mencionado, o *IDE* que utilizado foi o *Android Studio* com recurso à linguagem de programação *JAVA*. Através deste foi possível a utilização do *Surface View*, que permite a utilização da funcionalidade da câmara pela aplicação.

Para além da utilização destas tecnologias, também se vai fazer uso das funcionalidades disponibilizadas pela *Google Cloud Vision*, a ferramenta da *Google* responsável pelo reconhecimento de imagem - um dos pontos fulcrais no desenvolvimento da aplicação.

Nesta secção vão ser evidenciados e demonstrados os passos principais do seu processo de configuração.

De forma a tornar possível a utilização da aplicação por pessoas cegas e amblíopes, foi necessária a ativação e utilização da funcionalidade de *TalkBack*. Esta funcionalidade faz parte das ferramentas de acessibilidade disponibilizadas pelo *Android* e permite a utilização do aparelho sem ser necessário estabelecer contacto visual com o mesmo, uma vez que narra os textos que são apresentados no ecrã para o utilizador [20].

Nesta secção será detalhado o processo de integração da *Google Cloud Vision* com a aplicação assim como os principais passos que o tornaram possível, bem como a sua configuração. Para além disto também vai ser apresentado o protótipo final resultante de todo o processo descrito neste documento.

5.3.2 Configuração da Google Cloud Vision API

Para se iniciar o processo de integração da app com o Google Cloud Vision foram seguidos os seguintes passos:

- Criar uma conta na plataforma do Google Cloud [19]
- Criar um projeto na consola do Google Cloud [18]
- Instalar o Android Studio

Após estes passos, é possível a utilização da API da Google Cloud Vision na aplicação Android, após sido criada a conta na plataforma e criado o projeto na consola da Google Cloud Vision.

O passo seguinte foca-se na criação de uma chave da API para o nosso projeto.

Após clicarmos na opção *API key* aparece a chave criada.

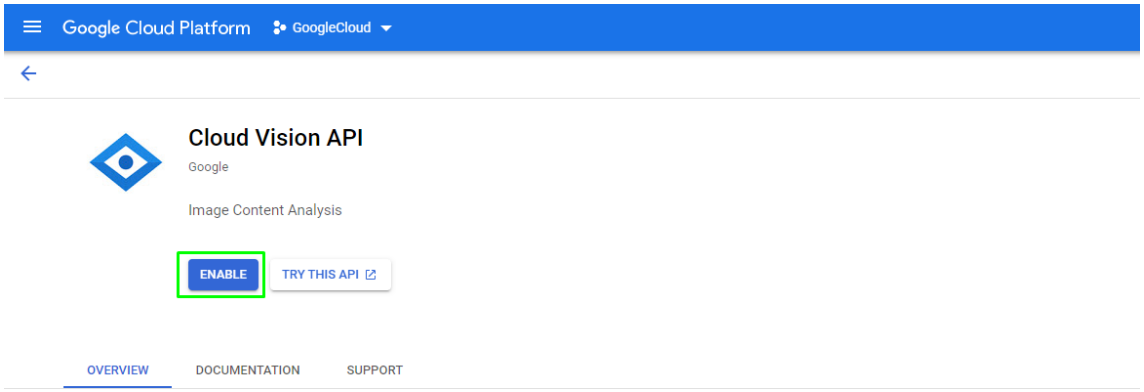

Figura 5.2: Activação do Google Cloud API

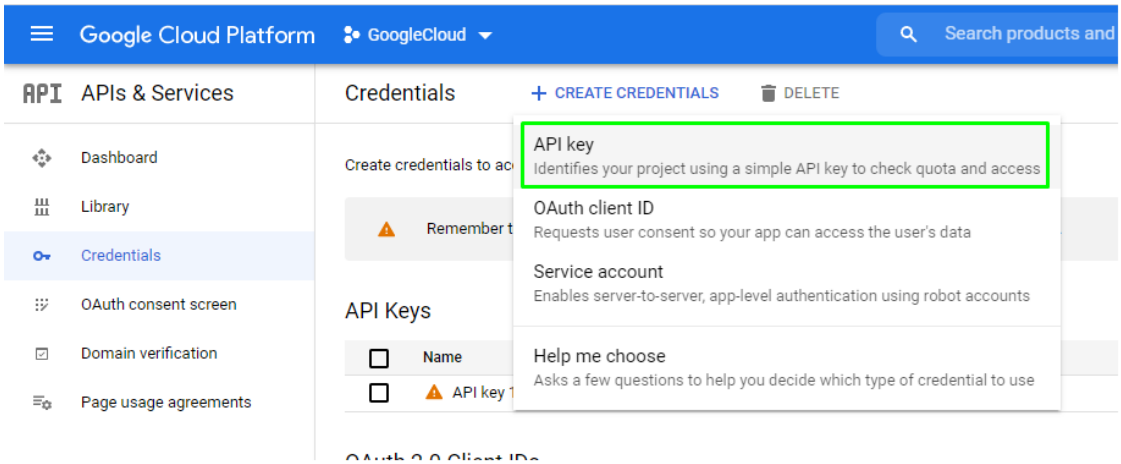

Figura 5.3: Criação da chave

Para aceder ao Cloud Vision API através do Android Studio, devemos adicionar as seguintes dependências ao ficheiro **build.gradle** do módulo app.

```
compile 'com.google.api-client:google-api-client-android:1.31.2' exclude module: 'httpclient'
compile 'com.google.code.findbugs:jsr305:2.0.1'
compile 'com.google.apis:google-api-services-vision:v1-rev451-1.25.0' exclude module: 'httpclient'
```

Figura 5.4: Dependências

Dado que o Google API Client apenas funciona se tivermos a permissão da Internet, é necessário ter a seguinte linha de código presente no ficheiro AndroidManifest.xml.

O cliente da *Google API* deve ser configurado antes de ser usado de forma a poder

```
<uses-permission android:name="android.permission.INTERNET"/>
```

Figura 5.5: Permissão à Internet

interagir com a *Cloud Vision API*. Os principais passos desse processo envolvem especificar a chave da *API*, o *HTTP transport* e a *JSON factory* desejados.

Como os nomes indicam, o *HTTP transport* vai ser responsável por comunicar com os servidores da *Google* e a *JSON factory*, entre outras coisas, vai ter como encargo a conversão de resultados em formato *JSON* em objetos *Java*.

A classe *Vision* representa o cliente da *Google API* na *Cloud Vision API*. Nesta classe existe uma outra classe, *Vision.Builder* que permite instanciar o cliente de uma forma mais simples.

Ao utiliza-la deve ser chamado o método *setVisionRequestInitializer()* onde é especificada a chave da *API*, como é mostrado na figura 5.6.

```
VisionRequestInitializer requestInitializer = new VisionRequestInitializer(CLOUD_VISION_API_KEY);

Vision.Builder builder = new Vision.Builder
        (httpTransport, jsonFactory,  httpRequestInitializer: null);

builder.setVisionRequestInitializer(requestInitializer);

Vision vision = builder.build();
```

Figura 5.6: Configuração da classe *Vision Builder*

Tendo esta configuração inicial feita, o processo de desenvolvimento pôde ser iniciado.

5.3.3 Layout

Na imagem 5.7 está representado o layout da aplicação. É usado o componente *SurfaceView* que é onde se encontra a área principal da aplicação. Este é usado para exibir o vídeo em tempo real.

O *layout* da aplicação foi feito de modo a que, quando se inicia a aplicação, seja apresentada a imagem que está a ser capturada pela câmara. Esta imagem será processada em tempo real e os resultados do reconhecimento feito pela *Google Cloud Vision API* são

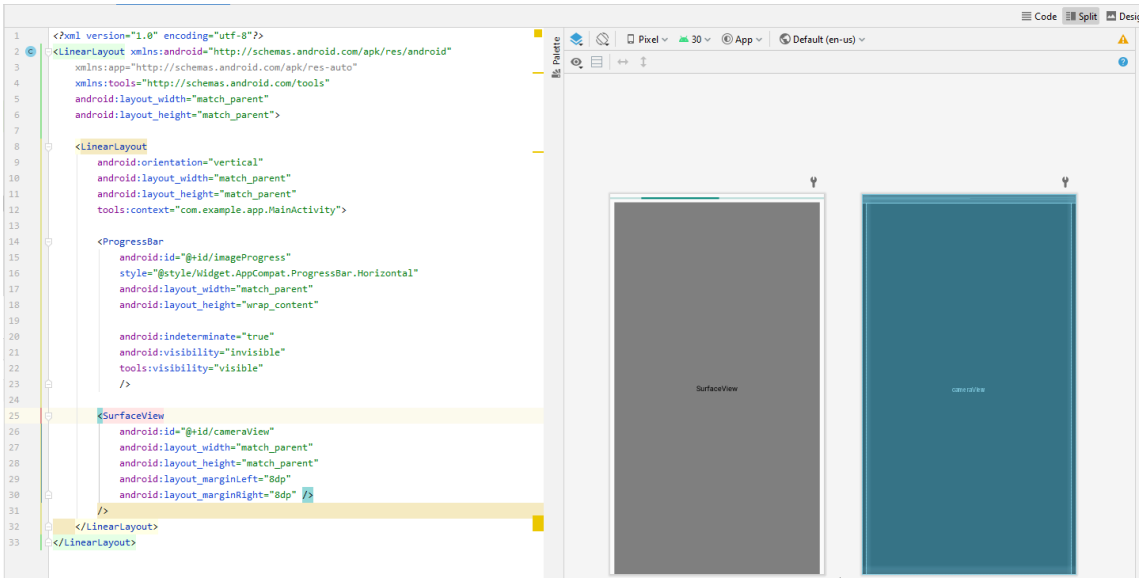

Figura 5.7: Layout

apresentados ao utilizador no ecrã.

Tendo o *layout* definido, é necessário fazer a integração da câmara, processo que será explicado na próxima subsecção.

5.3.4 Integração da câmara

Usando o componente *Surface-View*, exemplificado na imagem 5.7, são apresentadas as imagens que estão a ser capturadas pela câmara.

Para permitir a pré-visualização da captura da câmara foi criado o método *setupCameraPreview*.

Este método vai obter permissões para utilizar a câmara, abrir uma conexão com a mesma, inicializar o componente *Surface-View*, referido anteriormente, e criar uma sessão de captura e depois configurar um pedido de captura de imagem.

Com a captura de imagens funcional, estas vão ter de ser enviadas para a *Google Cloud Vision API* de forma a ser feito o seu reconhecimento.

5.3.5 Integração da *Google Cloud Vision API* com a aplicação

Nesta subsecção vai ser explicada como é feito o reconhecimento das imagens recebidas pela *Google Cloud Vision*. Para isso serão apresentados alguns excertos de código referentes

```
private void setupCameraPreview() {
    preview=(SurfaceView)findViewById(R.id.cameraView);
    preview.requestFocus();
    previewHolder=preview.getHolder();
    previewHolder.addCallback(surfaceCallback);
    previewHolder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);
}
```

Figura 5.8: Função setupCameraPreview()

a este processo.

A imagem capturada pela câmara vai ser enviada para o método *onActivityResult* da classe *Activity* que permitirá o acesso a um objeto *Bundle* que vai conter toda a informação presente na imagem enviada. Com este objeto é possível converter a informação da imagem noutros objetos, nomeadamente num objeto da classe *Bitmap*.

```
1  @Override
2     protected void onActivityResult(int requestCode, int resultCode
          , Intent data) {
3          super.onActivityResult(requestCode, resultCode, data);
4          if (requestCode == CAMERA_REQUEST_CODE && resultCode ==
              RESULT_OK) {
5              bitmap = (Bitmap) data.getExtras().get("data");
6              callCloudVision(bitmap, feature);
7          }
8     }
```

Listing 5.1: Função onActivityResult

O pedido que será feito à *Cloud Vision API* tem de ser feito com uma *string Base-64* que contém a informação da imagem. Como tal, é impossível enviar um objeto *Bitmap* diretamente. Então, foi a criada a função *getImageEncodeImage(Bitmap bitmap)* que permite a transformação de um objeto *Bitmap* numa *string Base-64*.

Esta função faz uso do método *compress()* da classe *Bitmap* que vai receber como argumentos o formato de compressão da imagem, a qualidade desejada do *output* e um objeto *ByteArrayOutputStream*.

No caso da função desenvolvida, a conversão foi feita utilizando o formato de imagem *JPEG*.

```
1  private Image getImageEncodeImage(Bitmap bitmap) {
2          Image base64EncodedImage = new Image();
3          ByteArrayOutputStream byteArrayOutputStream = new
               ByteArrayOutputStream();
4          long startTime = System.nanoTime();
5          bitmap.compress(Bitmap.CompressFormat.JPEG, 35,
               byteArrayOutputStream);
6          long endTime = System.nanoTime();
7          long methodeDuration = (endTime - startTime) / 1000000;
8          Log.i(TAG, "getImageEncodeImage " + methodeDuration + "
               milissegundos");
9
10         byte[] imageBytes = byteArrayOutputStream.toByteArray();
11         System.out.println(imageBytes);
12
13         // Base64 encode the JPEG
14         base64EncodedImage.encodeContent(imageBytes);
15
16         return base64EncodedImage;
17     }
```

Listing 5.2: Função getImageEncodeImage

Um objeto da classe *Feature* indicará qual o tipo de deteção de imagem que será feita, reconhecimento de texto, reconhecimento de logótipo ou reconhecimento de pontos de referência. Para além disto também indicará o número de resultados com as melhores pontuações a devolver, que no caso desta aplicação está definido como 10.

A seguir está demonstrado o método *buildFeature* que faz a construção deste objeto recebendo como argumentos de entrada o tipo de deteção de imagem que será feita e o número máximo de resultados.

```
1  private Feature buildFeature(String type, int maxResults){
2          Feature feature = new Feature();
3          feature.setType(type);
```

```
        feature.setMaxResults(maxResults);

        return feature;
    }
```

Listing 5.3: Função buildFeature

Neste ponto foi criado o pedido que será feito à *Cloud Vision API* com os métodos que foram descritos anteriormente.

```
        AnnotateImageRequest annotateImageReq = new
            AnnotateImageRequest();
        annotateImageReq.setFeatures(featureList);
        annotateImageReq.setImage(getImageEncodeImage(bitmap));
        annotateImageRequests.add(annotateImageReq);
```

Listing 5.4: Criação do pedido

A seguir está apresentada a interação com a *Cloud Vision API*, onde é configurada a conexão à *API*, configurado o pedido descrito no parágrafo anterior, enviado o pedido e devolvida a resposta recebida convertida para *string*.

```
new AsyncTask<Object, Void, String>() {
            @RequiresApi(api = Build.VERSION_CODES.N)
            @Override
            protected String doInBackground(Object... params) {
                try {

                    @SuppressLint("StaticFieldLeak") HttpTransport
                        httpTransport = AndroidHttp.
                        newCompatibleTransport();
                    JsonFactory jsonFactory = GsonFactory.
                        getDefaultInstance();

                    VisionRequestInitializer requestInitializer =
                        new VisionRequestInitializer(
                        CLOUD_VISION_API_KEY);

```

```
12                  Vision.Builder builder = new Vision.Builder
13                          (httpTransport, jsonFactory, null);
14
15                  builder.setVisionRequestInitializer(
                        requestInitializer);
16
17                  Vision vision = builder.build();
18
19                  BatchAnnotateImagesRequest
                        batchAnnotateImagesRequest = new
                        BatchAnnotateImagesRequest();
20                  batchAnnotateImagesRequest.setRequests(
                        annotateImageRequests);
21
22                  Vision.Images.Annotate annotateRequest = vision
                        .images().annotate(
                        batchAnnotateImagesRequest);
23                  annotateRequest.setDisableGZipContent(true);
24                  long startTime = System.nanoTime();
25                  BatchAnnotateImagesResponse response =
                        annotateRequest.execute();
26                  long endTime = System.nanoTime();
27                  long methodeDuration = (endTime - startTime) /
                        1000000;
28                  Log.i(TAG, "callCloudVision " + methodeDuration
                         + "milissegundos");
29
30                  return convertResponseToString(response);
31              } catch (GoogleJsonResponseException e) {
32                  Log.d(TAG, "failed to make API request because
                        " + e.getContent());
33              } catch (IOException e) {
34                  Log.d(TAG, "failed to make API request because
                        of other IOException " + e.getMessage());
35              }
```

```
36                  return "Cloud Vision API request failed. Check logs
                        for details.";
37              }
```

Listing 5.5: Processamento da imagem

5.3.6 Protótipo final

Nesta subsecção irá ser apresentado o protótipo final deste projeto.

Em geral, a Google Cloud Vision API suporta muitos tipos de técnicas de análise de imagem.

Irá ser demonstrado o reconhecimento de texto, deteção de pontos de referência e a deteção de logótipos.

Na figura 5.9 é demonstrada a técnica de reconhecer logótipos, neste caso reconheceu o *McDonald's* e é mostrada a sua precisão.

Figura 5.9: Reconhecimento de Logotipo

Na figura 5.10 é demonstrada a técnica de reconhecimento de pontos de referência, neste caso reconheceu a Avenida Central do Jardim e é mostrada a sua precisão.

Na figura 5.11 é demonstrada a técnica de reconhecimento de texto, no qual nos permite extrair texto de uma imagem. Neste caso em específico foi reconhecido o banco *Santander*.

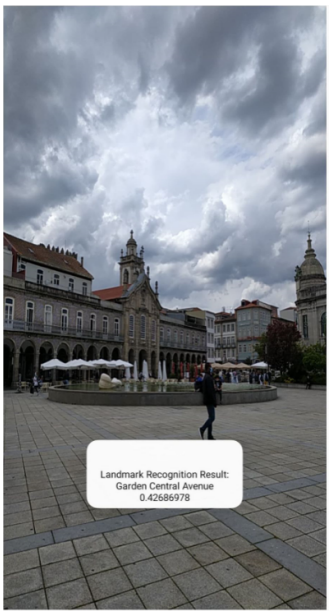

Figura 5.10: Reconhecimento de um ponto de referência

Figura 5.11: Reconhecimento de texto

Capítulo 6

Avaliação

Este capítulo começa por apresentar a metodologia utilizada para a realização dos testes e depois apresenta a avaliação da solução a partir de duas perspetivas diferentes. Primeiro, são apresentados os resultados da avaliação individual do reconhecimento do texto, logótipo e pontos de referência e, numa segunda fase, são apresentados os resultados dos testes realizados num ambiente real. Os resultados são devolvidos através da ferramenta de acessibilidade, Talkback.

6.1 Metodologia

Os testes individuais realizados sobre o reconhecimento de texto, logótipo e pontos de referência visam medir o tempo de reconhecimento e a exatidão obtida.

Todos os testes foram efetuados com o Redmi Note 8 que está a correr na versão do Android - *10 QKQ1.200114.002* e a versão do MIUI - *Miui Global 12.0.3*.

O telemóvel tem a aplicação instalada e para a monitorização de testes, é usada o Android Debug Bridge (ADB), que é uma ferramenta de linha de comando versátil que permite a comunicação com uma instância de emulador ou com um dispositivo Android conectado via wi-fi.

Os testes individuais, e como resultado da situação de confinamento devido à COVID-19, foram realizados utilizando fotografias impressas com uma resolução de 3840x2160. Os testes seguiram a seguinte metodologia:

- A câmara embebida na app aponta para a imagem e aguarda o resultado do reco-

nhecimento;

- Os resultados (tempo de processamento e precisão) são armazenados
- A câmara embebida na app aponta para um local neutro, onde um resultado de *Image not recognized* é devolvido.
- O processo é repetido 20 vezes.

6.1.1 Reconhecimento do logotipo

Para a avaliação do reconhecimento do logótipo, foram escolhidas três imagens: McDonald's, Starbucks e Burger King mostrados na figura 6.1.

Figura 6.1: Logótipos usados para o reconhecimento

Para a realização dos testes, e numa primeira fase fruto da situação de confinamento devido ao COVID-19 foram imprimidos os logótipos com a dimensão 3840x2160. De forma a avaliar o impacto da resolução da captura dos logótipos, foi imprimido também um logotipo do McDonald's com a dimensão 1036x1024. Este teste tem como objetivo monitorizar a precisão e tempo de resposta do sistema de reconhecimento desenvolvido.

Os resultados obtidos em termos de tempo de processamento e precisão são ilustrados na Figura 6.2.

A sua análise permite verificar uma diferença significativa na precisão de reconhecimento do logótipo McDonald's de baixa resolução (apresentado como Mc (low-res)) em relação a todos os outros logótipos. Outra conclusão é que existem três medições nos quatro logótipos em que há um tempo de processamento de aproximadamente 4.500 milis-

	Burger King	McDonald's 3840x2160	McDonald's 1036x1024	Starbucks
Precisão média	0,99	0,98	0,79	0,98
Tempo médio (milisegundos)	3457,47	3605,78	3658,10	3605,78

Tabela 6.1: Tempo médio e precisão média de cada logotipo

segundos. Este valor corresponde ao primeiro pedido feito à API, e após pedidos contínuos este valor tende a diminuir.

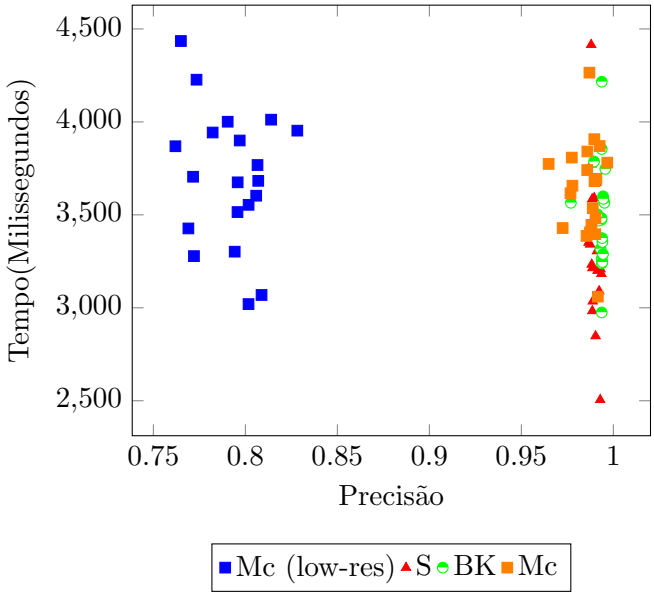

Figura 6.2: Avaliação dos resultados obtidos nos logótipos

A tabela 6.1 indica os valores médios dos tempos de processamento e precisão de reconhecimento para cada um dos logótipos. Os valores obtidos são muito semelhantes para os logótipos de maior resolução; o único valor a destacar é a menor exactidão do reconhecimento do logótipo com uma resolução inferior.

6.1.2 Reconhecimento de texto

Para a avaliação do reconhecimento de texto, foram escolhidas três imagens de lugares com texto e são apresentadas na Figura 6.3.

A primeira da esquerda representa uma pizzaria e o texto nela contido é "MAMMA MIA Ristorante Pizzeria; a segunda representa um nome de rua e em letra menor o código postal e o texto completo é "4710-079 Rua José Antunes Guimarães Gualtar"; a terceira imagem é o nome de uma loja de roupa e o texto é "ARRANJOS DE ROUPA D.AMÉLIA".

Figura 6.3: Imagens usadas para a avaliação do reconhecimento de texto. Da esquerda para a direita: (a) MAMMA MIA Ristorante Pizzeria; (b) 4710-079 Rua José Antunes Guimarães Gualtar; (c) ARRANJOS DE ROUPA D.AMÉLIA

$$accuracy = \frac{\#(character/word \ \times \ correctly\ recognized)}{\#(character/word \ \times \ recognized)}$$

Figura 6.4: Formula da precisão usada para reconhecimento de texto

O teste realizado segue o mesmo procedimento explicado na secção Metodologia.

Nesta prova específica, para cada imagem, foram efetuados testes frontais, laterais e à distância para compreender o impacto de cada um destes factores na precisão do algoritmo de reconhecimento. Para medir a precisão, utilizámos a fórmula de precisão apresentada na Figura 6.4 [39].

Ângulo frontal

No ângulo frontal, todos os testes efetuados apresentaram uma precisão de 100% como mostrado na figura 6.5.

Na tabela 6.2 é possível verificar um tempo de processamento ligeiramente mais longo na figura 6.3 (b), cuja razão é entendida porque há mais letras nesta amostra específica e algumas delas com pequenas dimensões.

Ângulo lateral

Na figura 6.6, são apresentados o tempo de processamento do reconhecimento do texto de cada imagem, bem como a exatidão, de acordo com a fórmula indicada na Figura 6.4.

	Fig 6.3 (a)	Fig 6.3 (b)	Fig 6.3 (c)
Precisão média	1	1	1
Tempo médio (milisegundos)	3927,4	4572,7	3803,25

Tabela 6.2: Tempo médio de processamento e precisão média do reconhecimento de texto de ângulo frontal

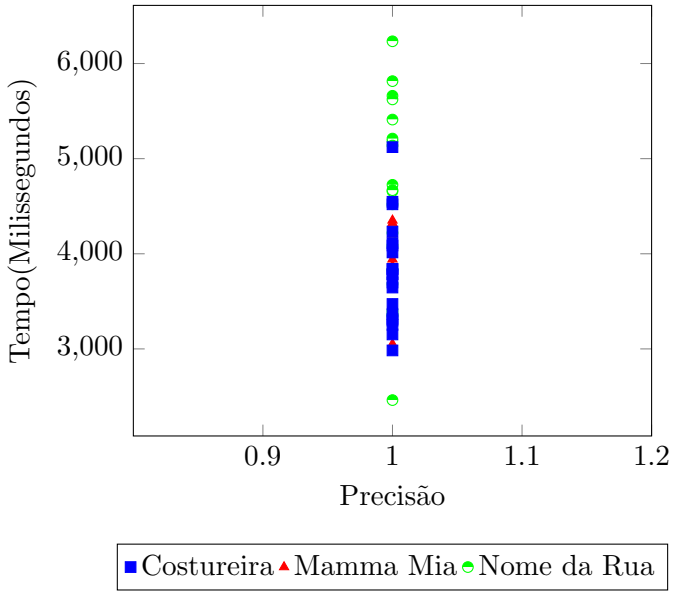

Figura 6.5: Resultados obtidos no ângulo frontal

O reconhecimento do texto lateral foi pior na figura 6.3 (b), devido ao pequeno tamanho das cartas relacionadas com o código postal.

No caso da figura 6.3 (a), nos 20 testes que foram realizados, apenas em 3 destes, a letra "N"da palavra "RISTORANTE"foi reconhecida corretamente, apresentado como letra "M".

No caso da figura 6.3 (b), nesses 20 testes, por 6 vezes não foi reconhecido corretamente o código postal (4710-079). No caso da figura 6.3 (c), em dois dos testes, a letra "D."não foi reconhecida corretamente.

Na tabela 6.3 e nos resultados apresentado na figura 6.6 é possível perceber que o reconhecimento da precisão média do texto da figura 6.3 (b) era o mais baixo, e o tempo de processamento era também o mais alto.

	Fig 6.3 (a)	Fig 6.3 (b)	Fig 6.3 (c)
Precisão média	0,99	0,93	0,99
Tempo médio (milisegundos)	3849,8	4235,25	4011,05

Tabela 6.3: Tempo médio de processamento e precisão média do reconhecimento de texto de ângulo lateral

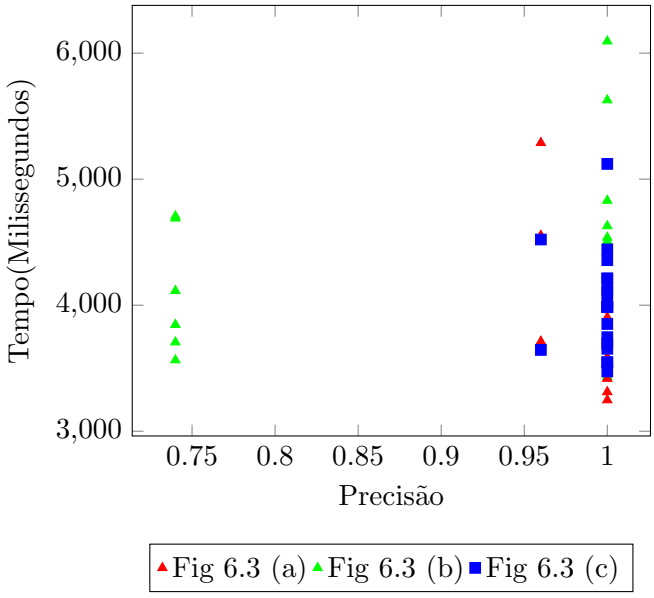

Figura 6.6: Resultados de reconhecimento de ângulo lateral

Testes de um ângulo mais distante

Para a realização deste testes, usamos um reconhecimento de imagem de três metros de distância. Na figura 6.7, existe alguma heterogeneidade nos resultados obtidos.

No caso da figura 6.3 (a), em 4 dos testes realizados, a letra "R"da palavra "Ristorante"não foi reconhecida corretamente. O algoritmo reconheceu como sendo a letra "Q".

No caso da figura 6.3 (b), o código postal não foi reconhecido 13 em 20 das vezes, devido ao tamanho pequeno das letras e em 4 casos as palavras também não foram identificadas corretamente.

No caso da figura 6.3 (c), as palavras "D.Amélia"nunca foram devidamente reconhecidas e isto deveu-se ao facto de haver uma luminosidade diferente na parte de cima da imagem capturada.

Na Tabela 6.4 é possível ver o tempo médio de processamento e precisão média para o reconhecimento do texto nas três imagens, que são coerentes com os resultados apresentados na Figura 6.7.

	Fig 6.3 (a)	Fig 6.3 (b)	Fig 6.3 (c)
Precisão média	0,99	0,72	0,69
Tempo médio (ms)	3564,15	3673,55	3508,65

Tabela 6.4: Tempo Médio de Processamento e Precisão Média por Distância

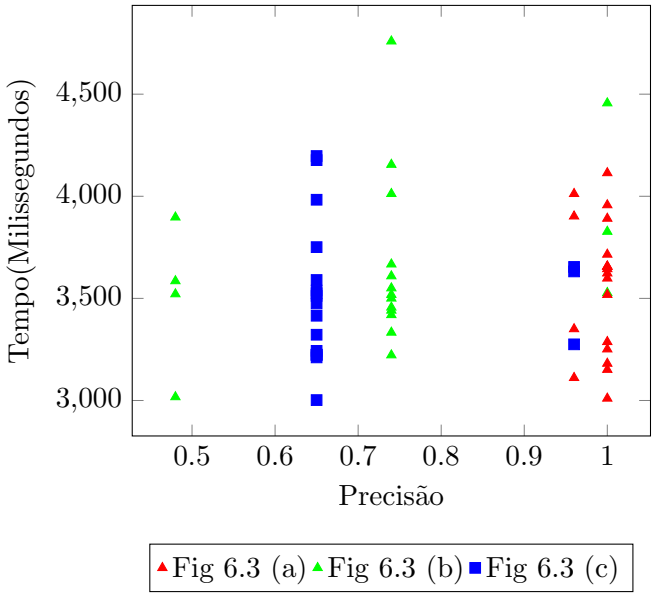

Figura 6.7: Resultados do reconhecimento de texto de 3 metros de distância

6.1.3 Reconhecimento de pontos de referência

Para a avaliação do reconhecimento de pontos de referência, três pontos de referência ilustrados na figura 6.8 foram escolhidos: (a) Bom Jesus, na cidade de Braga, (b) Santa Luzia, na cidade de Viana do Castelo e (c) Avenida Central, na cidade de Braga.

O Google Cloud Vision API retorna uma pontuação de confiança associada ao reconhecimento de pontos de referência, que foi utilizada na Figura 6.9, onde é apresentada a pontuação para o reconhecimento das três imagens. Nunca houve uma pontuação acima de 0,9 e existem valores próximos de 0,65, mas em todas elas o texto devolvido foi o correto.

A pontuação não está, portanto, diretamente relacionada com a identificação correta do ponto de referência.

Para eliminar variáveis como resolução, brilho e outros factores que poderiam ser associados a estes valores mais baixos na pontuação de reconhecimento, foi feito um teste com uma imagem de "Bom Jesus"diretamente na consola web API do Google e a pontuação foi de 0,8, o que indica que estes são os valores médios devolvidos pela API para a identificação de pontos de referência. Note-se que em todos os testes realizados para os 3 pontos de referência, e embora as pontuações sejam as mostradas na Figura 6.9, a localização foi sempre corretamente identificada em termos do texto devolvido.

Figura 6.8: Imagens usadas para o reconhecimento de pontos de referência: (a) Bom Jesus, na cidade de Braga, (b) Santa Luzia, na cidade de Viana do Castelo e (c) Avenida Central, na cidade de Braga.

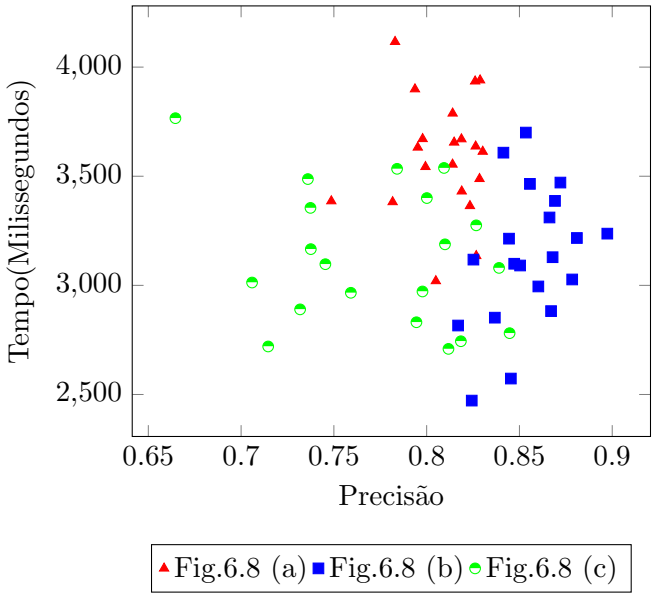

Figura 6.9: Resultados de reconhecimento de pontos de referência

A tabela 6.5 apresenta o tempo de processamento e a precisão do reconhecimento de pontos de referência. Como mencionado anteriormente, os valores médios da precisão são normais para o reconhecimento de um ponto de referência utilizando a API do Google Cloud Vision e o local foi sempre corretamente identificado.

	Fig.6.8 (a)	Fig.6.8 (b)	Fig.6.8 (c)
Precisão média	0,81	0,86	0,78
Tempo médio (ms)	3592,95	3133,2	3125,6

Tabela 6.5: Tempo Médio de Processamento e Precisão Média de Reconhecimento de pontos de referência

6.1.4 Testes em cenário real

O objetivo principal desta subsecção é mostrar o funcionamento real da aplicação em teste num ambiente real.

Para além dos testes individuais de logótipos, textos e pontos de referência apresentados nas secções anteriores, foi concebido um percurso num local central da cidade de Braga, em Portugal, de modo a poder ser realizado um verdadeiro cenário de teste no terreno, representando um percurso que uma pessoa poderia realizar numa situação real, passando por locais dos três tipos considerados.

A figura 6.10 representa o local onde os testes foram realizados, com dois tipos de pontos a serem identificados:

1. lugares a serem reconhecidos (representados com cor vermelha). Foram escolhidos quatro lugares: (1) - McDonald's (Logotipo), (2) - Banco Santander (Texto), (3) - Avenida Central do Jardim (Ponto de referência) e (4) - Jupial (Texto)

2. lugares de onde o reconhecimento foi feito (representados na cor preta)

Figura 6.10: Cenário do mapa dos testes de campo mostrando os lugares a serem reconhecidos (1 a 4) e os lugares onde o reconhecimento foi feito (5 a 11)

O seguinte percurso foi realizado cinco vezes: (5) - (9) - (6) - (7) - (10) - (11) - (8). A

6.6 mostra os lugares onde foi feito o reconhecimento de cada um dos quatro lugares. Os locais de texto (1 e 4) só foram reconhecidos a partir de um local e a uma curta distância. O logótipo (1) e o ponto de referência (3) foram reconhecidos em mais lugares, visto que são maiores e mais visíveis à distância.

Lugares de onde o reconhecimento é feito	Lugares a reconhecer
(5) (9)	(1)
(6)	(2)
(7) (10) (11)	(3)
(8)	(4)

Tabela 6.6: Correspondência entre os locais a reconhecer e os locais onde o reconhecimento será feito

A tabela 6.7 representa a precisão e o tempo de processamento do reconhecimento do texto dos locais (1) e (4). Nos 5 percursos realizados como teste, os textos dos dois locais foram corretamente identificados em todas as situações, com uma exatidão de 100%, de acordo com a fórmula definida na Figura 6.4.

	Santander	Jupial
Precisão média	1	1
Tempo médio (ms)	1694,4	1852,2

Tabela 6.7: Tempo Médio de Processamento e Precisão Média do Reconhecimento de texto em testes no mundo real

A tabela 6.8 representa a precisão e o tempo de processamento do reconhecimento do logótipo (1) a partir de dois locais diferentes, um mais próximo (5) e outro com alguma distância (9). Nos 5 percursos realizados como teste, o logótipo foi reconhecido com uma precisão de 97% a partir do local mais distante e com uma precisão de 98% a partir do local mais próximo.

	Desde local (5)	Desde local (9)
Precisão média	0,98	0,97
Tempo médio (ms)	1687,4	1605

Tabela 6.8: Tempo Médio de Processamento e Precisão Média do Reconhecimento de Logotipo em testes no mundo real

A tabela 6.9 representa a precisão e o tempo de processamento do reconhecimento do ponto de referência (3) a partir de três locais diferentes.

A partir do local mais próximo (7), o ponto de referência foi reconhecido com uma

precisão de 49%, o que se deve ao facto de o ponto de referência não ser totalmente capturado pela câmara devido à sua proximidade do local.

A partir do local (11), havia um chafariz em frente do ponto de referência, o que também reduziu a precisão do reconhecimento.

O reconhecimento a partir do local (10) foi superior aos outros, dado que a imagem capturada não tem quaisquer obstáculos significativos e o ponto de referência pode ser capturado na sua totalidade. Apesar da precisão do reconhecimento, em duas situações, valores presentes em torno de 50%, o nome do ponto de referência devolvido ao utilizador foi sempre correto.

	Desde o local (7)	Desde o local (10)	Desde o local (11)
Precisão média	0,49	0,64	0,48
Tempo médio (ms)	2196,8	1976,6	2278,6

Tabela 6.9: Tempo Médio de Processamento e Precisão Média de Reconhecimento de Pontos de Referência no cenário real

Capítulo 7

Conclusão e trabalho futuro

Ao longo deste documento, foi proposta, implementada e testada uma aplicação móvel com o objetivo final de ajudar o segmento de pessoas com deficiência visual a deslocar-se nas cidades e obter informações sobre a sua localização atual e assim ajudar em situações em que perdem a sua orientação.

A solução utiliza a abordagem de Posicionamento Visual e a API do Google Cloud Vision para reconhecer imagens, nomeadamente texto, logótipos e pontos de referência.

A solução foi avaliada individualmente nos três aspetos utilizando fotografias, e foi também realizado um teste real integrador na cidade de Braga.

No que diz respeito ao reconhecimento de logótipos, foi alcançada uma precisão de 98% com o reconhecimento utilizando fotografias com uma resolução de 3840x2160. A diminuição da resolução da imagem para 1036x1024 teve um impacto considerável na precisão, que neste caso foi de 79%. O tempo médio de processamento foi de 3,5 segundos, utilizando uma rede Wi-Fi.

Em relação ao reconhecimento de texto, foram utilizadas três imagens diferentes, e foram consideradas três posições: frontal, lateral e com 3 metros de distância. No ângulo frontal, a precisão do reconhecimento foi de 100%. No ângulo lateral, foi alcançada uma precisão de 99% em duas das imagens e uma precisão de 93% noutra, o que se justifica pelo facto de ser uma imagem com letras mais pequenas e que não será tão visível de um ângulo lateral. Finalmente, a uma distância de 3 metros, a precisão foi visivelmente menor, com muitas letras a serem incorrectamente identificadas. Neste caso, o valor médio de exactidão foi de 80%. Em relação aos tempos de processamento, a média foi de 4,4

segundos, sendo os valores mais elevados registados no reconhecimento da imagem com letras mais pequenas e em maior número.

O reconhecimento de um ponto de referência obteve uma precisão média de 82%, o que é considerado um bom valor, pois este é o valor médio obtido quando uma fotografia nítida e de boa resolução é submetida para reconhecimento diretamente na consola web da API. Mesmo tendo em conta o valor de precisão, o resultado devolvido na identificação do ponto de referência foi sempre correto. O tempo médio foi de 3,2 segundos.

Os testes realizados em cenário real foram realizados na cidade de Braga. Para este efeito, foi concebido um percurso que incluía o reconhecimento de dois textos, um ponto de referência e um logotipo. A precisão do reconhecimento do texto foi de 100% com um tempo médio de processamento de 1,8 segundos. O reconhecimento de logótipos tinha uma precisão de 98%, e a maior distância ao logótipo não era significativa. A média de processamento, neste caso, foi de 1,6 segundos. Finalmente, o reconhecimento do ponto de referência deu sempre a localização correcta como um retorno, embora a precisão variasse quando estava demasiado perto e quando havia obstáculos à frente. O tempo médio de processamento, neste caso, foi de 2,2 segundos. A diminuição do tempo médio de processamento em testes realizados num cenário real é realçada,o que pode ser eventualmente justificado pela utilização de dados móveis.

A utilização desta API revelou-se adequada para a investigação do problema definido, e pode ser uma solução viável para incorporação numa aplicação móvel com o objetivo de ajudar os deficientes visuais a terem uma maior orientação quando se deslocam nas cidades.

No âmbito de trabalho realizado na tese foi publicado o artigo na conferência *IEEE International Smart Cities Conference 2021* intitulado *Inclusive Mobility Solution for Visually Impaired People using Google Cloud Vision* [28].

Relativamente a possíveis desenvolvimentos ou expansões futuras, existe espaço para a utilização de um dataset mais extenso, capaz de reconhecer locais que ainda não são possíveis de ser reconhecidos pela API, e até incluir outros tipos de pontos de referência, como por exemplo parques e jardins, cujo reconhecimento requer um conjunto de dados maior.

Referências

[1] *About small cells – Small Cell Forum*. https://www.smallcellforum.org/small-cells/. (Accessed on 07/13/2021).

[2] ACAPO. *Deficiência visual*. Last accessed on May 24th 2021. 2021. URL: http://www.acapo.pt/deficiencia-visual/perguntas-e-respostas/deficiencia-visual#quantas-pessoas-com-deficiencia-visual-existem-em-portugal-202.

[3] ACAPO. *Orientação e mobilidade*. Last accessed on July 12th 2021. 2021. URL: https://www.acapo.pt/deficiencia-visual/perguntas-e-respostas/orientacao-e-mobilidade.

[4] Dragan Ahmetovic et al. "NavCog: A Navigational Cognitive Assistant for the Blind". Em: set. de 2016. DOI: 10.1145/2935334.2935361.

[5] ANACOM. *ACAPO - Associação dos Cegos e Amblíopes de Portugal*. Last accessed on May 24th 2021. 2002. URL: https://www.anacom.pt/render.jsp?categoryId=36666.

[6] Android. *Android — The platform pushing what's possible*. Last accessed on May 30th 2021. URL: https://www.android.com/.

[7] Desenvolvedores Android. *Arquitetura da plataforma*. Last accessed on May 24th 2021. 2010. URL: https://developer.android.com/guide/platform?hl=pt-br.

[8] Desenvolvedores Android. *Desenvolvedores Android — Android Developers*. Last accessed on May 30th 2021. URL: https://developer.android.com/.

[9] Desenvolvedores Android. *Desenvolver apps Android com o Kotlin*. Last accessed on May 30th 2021. URL: https://developer.android.com/kotlin.

[10] Microsoft Azure. *Computer Vision — Microsoft Azure*. URL: https://azure.microsoft.com/en-us/services/cognitive-services/computer-vision/.

[11] Anahid Basiri, Pouria Amirian e Adam Winstanley. "The use of quick response (QR) codes in landmark-based pedestrian navigation". Em: *International Journal of Navigation and Observation* 2014 (2014). ISSN: 16876008. DOI: 10.1155/2014/897103.

[12] Prashant Bhardwaj e Jaspal Singh. "Design and Development of Secure Navigation System for Visually Impaired People". Em: *International Journal of Computer Science and Information Technology* 5 (4 2013), pp. 159–164. ISSN: 09754660. DOI: 10.5121/ijcsit.2013.5413.

[13] BlindSquare. *BlindSquare*. Last accessed on July 13th 2021. URL: https://www.who.int/blindness/GLOBALDATAFINALforweb.pdf?ua=1.

[14] Hsuan-Eng Chen et al. "BlindNavi". Em: (Figure 1 2015), pp. 19–24. DOI: 10.1145/2702613.2726953.

[15] Hsuan-Eng Chen et al. "BlindNavi: A Navigation App for the Visually Impaired Smartphone User". Em: CHI EA '15 (2015), pp. 19–24. DOI: 10.1145/2702613.2726953. URL: https://doi.org/10.1145/2702613.2726953.

[16] Clarifai. *computer vision and api enterprise platform*. URL: https://www.clarifai.com.

[17] Google Cloud. *Documentação do Cloud Vision — API Cloud Vision — Google Cloud*. URL: https://cloud.google.com/vision/docs.

[18] Google Cloud. *Google Cloud Platform*. URL: https://console.cloud.google.com/.

[19] Google Cloud. *Serviços de computação em nuvem — Google Cloud*. URL: https://cloud.google.com/.

[20] Melissa Cruz Cossetti. *O que é o TalkBack?* Last accessed on May 24th 2021. 2018. URL: https://tecnoblog.net/247247/o-que-e-o-talkback//.

[21] *Deficiência visual — Associação dos Cegos e Amblíopes de Portugal*. https://www.acapo.pt/deficiencia-visual/perguntas-e-respostas/deficiencia-visual. (Accessed on 10/26/2021).

[22] IBM Developer. *Fundamentos da linguagem Java – IBM Developer*. Last accessed on June 16th 2021. URL: https://developer.ibm.com/br/tutorials/j-introtojava1/.

[23] Organisation for Economic Cooperation e Development. *Embracing Innovation in Government - Global Trends*. https://www.oecd.org/gov/innovative-government/embracing-innovation-in-government-poland.pdf. (Accessed on 07/13/2021).

[24] A. Jin Fukasawa e Kazusihge Magatani. "A navigation system for the visually impaired an intelligent white cane". Em: *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS* (2012), pp. 4760–4763. ISSN: 1557170X. DOI: 10.1109/EMBC.2012.6347031.

[25] Ross Girshick. *Fast R-CNN*. 2015. arXiv: 1504.08083 [cs.CV].

[26] Ross Girshick et al. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation". Em: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (nov. de 2013). DOI: 10.1109/CVPR.2014.81.

[27] *Glossário — ACAPO*. https://www.acapo.pt/deficiencia-visual/glossario. (Accessed on 10/26/2021).

[28] Joana Gonçalves e Sara Paiva. "Inclusive Mobility Solution for Visually Impaired People using Google Cloud Vision". Em: *2021 IEEE International Smart Cities Conference (ISC2)*. 2021, pp. 1–7. DOI: 10.1109/ISC253183.2021.9562892.

[29] Google. *Primeiros passos no Android com o Talkback*. Last accessed on May 24th 2021. 2010. URL: https://support.google.com/accessibility/android/answer/6283677?hl=pt-BR.

[30] *Google Home – Apps no Google Play*. https://play.google.com/store/apps/details?id=com.google.android.apps.chromecast.app&hl=pt_PT&gl=US. (Accessed on 07/17/2021).

[31] *How are the terms low vision, visually impaired, and blind defined? — DO-IT*. https://www.washington.edu/doit/how-are-terms-low-vision-visually-impaired-and-blind-defined. (Accessed on 10/26/2021).

[32] IBM. *IBM Cloud Docs*. URL: https://cloud.ibm.com/docs/visual-recognition.

[33] IBM. *Watson Visual Recognition*. URL: https://www.ibm.com/cloud/watson-visual-recognition.

[34] *iBus*. URL: http://www.stm.info/en/about/major_projects/major-bus-projects/ibus-real-time.

[35] Affan Idrees, Zahid Iqbal e Maria Ishfaq. "an efficient indoor navigation technique to find optimal route for blinds using qr codes". Em: (2015).

[36] Noman Islam, Zeeshan Islam e Nazia Noor. *A Survey on Optical Character Recognition System*. 2016.

[37] Hanen Jabnoun, Mohammad Abu Hashish e Faouzi Benzarti. "Mobile Assistive Application for Blind People in Indoor Navigation". Em: *The Impact of Digital Technologies on Public Health in Developed and Developing Countries*. Ed. por Mohamed Jmaiel et al. Cham: Springer International Publishing, 2020, pp. 395–403. ISBN: 978-3-030-51517-1.

[38] Aditi Jog e Prof.Shirish Halbe. "Object Tracking Using Camshift Algorithm in Open CV". Em: *International Journal of Scientific Research* 1 (jun. de 2012), pp. 37–39. DOI: 10.15373/22778179/NOV2012/13.

[39] Markus Junker, Rainer Hoch e Andreas Dengel. "On the Evaluation of Document Analysis Components by Recall, Precision, and Accuracy". Em: (abr. de 2000). DOI: 10.1109/ICDAR.1999.791887.

[40] Esteban Bayro Kaiser e Michael Lawo. "Wearable navigation system for the visually impaired and blind people". Em: *Proceedings - 2012 IEEE/ACIS 11th International Conference on Computer and Information Science, ICIS 2012* (1 2012), pp. 230–233. DOI: 10.1109/ICIS.2012.118.

[41] Lukasz Kaminski et al. “”VOICE MAPS- Portable, dedicated GIS for supporting the street navigation and self-dependent movement of the blind”. Em: *Proceedings of the 2010 2nd International Conference on Information Technology, ICIT 2010* (February 2014 2010), pp. 153–156.

[42] Antonio Alfredo Ferreira Loureiro et al. “Computação ubıqua ciente de contexto: Desafios e tendências”. Em: *27o Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuıdos* (2009), pp. 99–149.

[43] Somayya Madakam e R. Ramaswamy. “Smart Homes (Conceptual Views)”. Em: *2014 2nd International Symposium on Computational and Business Intelligence*. Dez. de 2014, pp. 63–66. DOI: 10.1109/ISCBI.2014.21.

[44] Shoichi Maeyama, Akihisa Ohya e Shin’ichi Yuta. “Outdoor navigation using natural landmarks by teaching-playback scheme”. Em: *IEEE International Conference on Intelligent Robots and Systems* 3 (1997), pp. 17–18. DOI: 10.1109/iros.1997.656801.

[45] *Maps That You Can Hear and Touch - Bloomberg*. https://www.bloomberg.com/news/articles/2015-01-15/maps-that-you-can-hear-and-touch. (Accessed on 11/17/2021).

[46] Bruno Mendes e Lígia Torres Silva. “Integração de um sistema de monotorização ambiental urbano numa smart city”. Em: 2016.

[47] Sivapong Nilwong et al. “Outdoor Landmark Detection for Real-World Localization using Faster R-CNN”. Em: *ACM International Conference Proceeding Series* (August 2020 2018), pp. 165–169. DOI: 10.1145/3284516.3284532.

[48] Halima Mansour Omar et al. “Integration of GPS and dead reckoning navigation system using moving horizon estimation”. Em: *Proceedings of 2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2016* (1 2016), pp. 553–556. DOI: 10.1109/ITNEC.2016.7560421.

[49] *Optical Character Recognition (OCR) - Overview and Use Cases - viso.ai*. https://viso.ai/computer-vision/optical-character-recognition-ocr/. (Accessed on 11/17/2021).

[50] *Optical Character Recognition With Google Cloud Vision API*. Last accessed on May 30th 2021. URL: https://medium.com/hackernoon/optical-character-recognition-with-google-cloud-vision-api-255bb8241235.

[51] Ken Peffers et al. "A design science research methodology for information systems research". Em: *Journal of management information systems* 24.3 (2007), pp. 45–77.

[52] Ken Peffers et al. "The design science research process: A model for producing and presenting information systems research". Em: *Proceedings of First International Conference on Design Science Research in Information Systems and Technology DESRIST* (fev. de 2006).

[53] Future Peterborough. *GEORGIE PHONE*. URL: http://www.futurepeterborough.com/project/georgie-phone/.

[54] Konstantinos Plataniotis Petros Spachos. *BLE Beacons in the Smart City: Applications, Challenges, and Research Opportunities*. https://arxiv.org/pdf/2102.08751.pdf. (Accessed on 07/13/2021).

[55] Martin Raubal e Stephan Winter. "Enriching wayfinding instructions with local landmarks". Em: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 2478 (January 2002 2002), pp. 243–259. ISSN: 16113349. DOI: 10.1007/3-540-45799-2_17.

[56] Amazon Rekognition. *Amazon Rekognition – Análise de imagem e vídeo – Amazon Web Services (AWS)*. Last accessed on May 24th 2021. URL: https://aws.amazon.com/pt/rekognition/.

[57] Abbas Riazi et al. "Outdoor difficulties experienced by a group of visually impaired Iranian people". Em: *Journal of Current Ophthalmology* 28 (2 2016), pp. 85–90. ISSN: 24522325. DOI: 10.1016/j.joco.2016.04.002. URL: http://dx.doi.org/10.1016/j.joco.2016.04.002.

[58] Wilson Sakpere, Michael Adeyeye-Oshin e Nhlanhla B.W. Mlitwa. "A state-of-the-art survey of indoor positioning and navigation systems and technologies". Em: *South African Computer Journal* 29 (3 2017), pp. 145–197. ISSN: 23137835. DOI: 10.18489/sacj.v29i3.452.

[59] Mohammad A. Salahuddin et al. “An efficient artificial landmark-based system for indoor and outdoor identification and localization”. Em: *IWCMC 2011 - 7th International Wireless Communications and Mobile Computing Conference* (2011), pp. 583–588. DOI: 10.1109/IWCMC.2011.5982598.

[60] Júlio Santos. *Google adiciona comandos de voz para o TalkBack*. Fev. de 2021. URL: https://newvoice.ai/2021/02/25/google-adiciona-comandos-de-voz-para-o-talkback/.

[61] *Sendero Group: The Seeing Eye GPS™ App for cell-enabled iOS devices*. http://www.senderogroup.com/products/SeeingEyeGPS/index.html. (Accessed on 07/13/2021).

[62] SENSO. *Orientação e mobilidade*. Last accessed on July 12th 2021. 2021. URL: https://sensocrv.com.br/corporativo/palestra/artigos/7-orientacao-emobilidade.

[63] M. Serrão et al. “Indoor localization and navigation for blind persons using visual landmarks and a GIS”. Em: *Procedia Computer Science* 14 (Dsai 2012), pp. 65–73. ISSN: 18770509. DOI: 10.1016/j.procs.2012.10.008. URL: http://dx.doi.org/10.1016/j.procs.2012.10.008.

[64] Saleh Shadi et al. “Outdoor navigation for visually impaired based on deep learning”. Em: *CEUR Workshop Proceedings* 2514 (2019), pp. 397–406. ISSN: 16130073.

[65] Rupali A Tanpure. “Advanced Voice Based Blind Stick with Voice Announcement of Obstacle Distance”. Em: 4 (8 2018), pp. 85–87.

[66] Md. Siddiqur Rahman Tanveer, M. M. A. Hashem e Md. Kowsar Hossain. “Android assistant EyeMate for blind and blind tracker”. Em: *2015 18th International Conference on Computer and Information Technology (ICCIT)*. 2015, pp. 266–271. DOI: 10.1109/ICCITechn.2015.7488080.

[67] B. Thylefors et al. “Global data on blindness.” Em: *Bulletin of the World Health Organization* 73.1 (1995), pp. 115–121.

[68] Sylvie Treuillet e Eric Royer. "outdoor/indoor vision-based locali- zation for blind pedestrian navigation assistance". Em: *International Journal of Image and Graphics* 10 (4 2010), pp. 481–496. ISSN: 02194678. DOI: 10.1142/S0219467810003937.

[69] *Using Landmarks for Orientation – Point 4 of 5 of the 5 Point Travel System – BLIND ON THE MOVE*. https://blindonthemove.com/2020/04/27/using-landmarks-for-orientation-point-4-of-5-of-the-5-point-travel-system/. (Accessed on 11/01/2021).

[70] *Wayfindr - Accessible Indoor Audio Navigation*. https://www.wayfindr.net/. (Accessed on 07/13/2021).

[71] WHO. *global data on visual impairments 2010*. Last accessed on May 24th 2021. 2010. URL: https://www.who.int/blindness/GLOBALDATAFINALforweb.pdf?ua=1.

[72] Kumar Yelamarthi et al. "RFID and GPS integrated navigation system for the visually impaired". Em: *Midwest Symposium on Circuits and Systems* (2010), pp. 1149–1152. ISSN: 15483746. DOI: 10.1109/MWSCAS.2010.5548863.

[73] Aurang Zeb, Sehat Ullah e Ihsan Rabbi. "Indoor vision-based auditory assistance for blind people in semi controlled environments". Em: *2014 4th International Conference on Image Processing Theory, Tools and Applications, IPTA 2014* (2015), pp. 0-5. DOI: 10.1109/IPTA.2014.7001996.