



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

**Desenvolvimento de aplicativo móvel para auxílio de
indivíduos com discromatopsia utilizando visão
computacional**

Felipe Spínola de A. Ramos

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientador
Prof. Dr. Flávio de Barros Vidal

Brasília
2018

Dedicatória

Dedico este trabalho à minha família e ao leitor, com expectativa deste trabalho ser útil para os seus estudos e trabalhos futuros.

Agradecimentos

Primeiramente agradeço a **Deus**, meu criador, por ter me capacitado para a elaboração deste trabalho, pois se não fosse por Ele nada disso seria possível. Por ter me guiado e me dado forças em todos os momentos e por ser um Deus presente na minha vida, me abençoando em todos os momentos.

Agradeço à minha família, representados aqui pelos meus pais, **José Inácio Ramos** e **Angela Spínola de Araújo Ramos**, que sempre me apoiaram durante todo o meu curso e principalmente durante a elaboração deste trabalho.

Agradeço à minha namorada e melhor amiga, **Ana Lídia Peixoto Monteiro Barbosa**, pela motivação, companheirismo e carinho que me ajudaram na realização deste trabalho.

Agradeço ao meu orientador **Flávio de Barros Vidal**, por todo ensinamento que me foi dado em suas aulas, bem como as ideias compartilhadas em reuniões e principalmente por acreditar e incentivar a realização deste trabalho, sempre me auxiliando e apoiando durante todas as dificuldades.

Agradeço aos meus amigos que me acompanharam durante toda a caminhada, tornando-a mais fácil e leve.

Agradeço de uma forma especial ao meu amigo **Marcelo Egidio Brasileiro do Vale**, que me auxiliou na parte gráfica deste trabalho.

Agradeço também a todos os professores que tive oportunidade de ser aluno, os quais agregaram conhecimento na minha vida acadêmica.

Resumo

Daltonismo é uma deficiência visual não tão rara, visto que aproximadamente 8% da população mundial masculina a possui, que dificulta o indivíduo na distinção de determinadas cores, de acordo com o tipo de daltonismo. Este trabalho propõe uma ferramenta que possa, além de auxiliar indivíduos daltônicos no seu cotidiano a identificar objetos que possuem alguma cor na qual o indivíduo não consiga distingui-la, permitir que qualquer indivíduo com acesso a esta ferramenta descubra se possui ou não daltonismo de uma maneira rápida e eficiente. É proposta, então, a criação de um aplicativo para dispositivos móveis, permitindo maior acessibilidade e visando uma fácil adaptação para futuramente ser utilizado em óculos inteligentes, sendo que o aplicativo é calibrado por meio das imagens do teste de daltonismo mais utilizado no mundo, o teste de Ishihara.

Palavras-chave: daltonismo, visão computacional, aplicativo móvel, processamento de imagem, teste de Ishihara

Abstract

Color blindness is a not so rare visual deficiency, since nearly 8% of the male world population is colorblind, which makes it difficult for the individual to distinguish certain colors, according to the type of color blindness. This work proposes a tool that can, in addition to help colorblind people in their daily lives to identify objects that have some color in which they can't distinguish it, allow any person with access to this tool to find out whether or not they have color blindness in a fast and efficient way. Then it is proposed to develop a mobile app, allowing greater accessibility and aiming for an easy adaptation to this methodology to be used, in the future, in smart glasses, being that the application is calibrated through the images of the most used color blindness test in the world, the Ishihara test.

Keywords: color blindness, computer vision, mobile app, Ishihara test

Sumário

1	Introdução	1
1.1	Contextualização	1
1.2	Objetivos	2
1.2.1	Objetivo Secundário	3
1.3	Distribuição do Manuscrito	3
2	Fundamentação Teórica	4
2.1	Daltonismo	4
2.1.1	Classificações de Daltonismo	5
2.1.2	Incidência dos Tipos	5
2.2	Teste de Ishihara	6
2.2.1	Funcionamento do Teste	6
2.2.2	Instruções do Teste	9
2.2.3	Análise dos Resultados do Teste	10
2.3	Visão Computacional	11
2.3.1	Espaços de Cores	11
2.3.2	Segmentação por Cor	13
2.4	OpenCV	14
2.5	Sistema <i>iOS</i> & Objective-C	15
3	Metodologia Proposta	16
3.1	Descrição Inicial da Solução Proposta	16
3.1.1	Motivação	17
3.1.2	Requisitos do Aplicativo	17
3.1.3	Modelagem do Sistema	18
3.1.4	Funcionalidades	21
3.2	Desenvolvimento da Aplicação Móvel	22
3.2.1	Fluxograma	23
3.2.2	Criação da Apresentação Inicial	24

3.2.3	Teste de Ishihara no Aplicativo	25
3.2.4	Telas com o Resultado do Teste	26
3.2.5	Importância da Determinação do Tipo de Discromatopsia	28
3.2.6	Estratégica de Identificação de Cores Problemáticas	29
3.2.7	Permissão de Acesso e Instruções de Uso	32
3.2.8	Utilização do <i>OpenCV</i>	33
3.2.9	Solução Utilizando a Câmera	34
4	Implementação da Solução	35
4.1	Configuração da Câmera	35
4.2	Processamento de Imagem	36
4.2.1	Mudança do Espaço de Cores	36
4.2.2	Transformações Morfológicas	37
4.2.3	Criação de Máscaras de Intervalos de Cores	38
4.2.4	Encontrar e Desenhar Contornos	41
4.3	Interação com o Usuário	42
5	Resultados	46
5.1	Obtenção dos Resultados	46
5.2	Filtro para Prancha de Número 2	47
5.3	Filtro para Pranchas de Números 3 a 5	48
5.4	Filtro para Pranchas de Números 6 a 9	48
5.5	Filtro para Pranchas de Números 10 a 13	49
5.6	Filtro para Pranchas de Números 14 e 17	50
5.7	Filtro para Pranchas de Números 15 e 16	51
5.8	Filtro para Pranchas de Números 22 a 25	52
6	Conclusões e Trabalhos Futuros	55
6.1	Trabalhos Futuros	56
	Referências	57

Lista de Figuras

2.1	Todos conseguem identificar o número 12. Retirado de [1].	7
2.2	Indivíduos normais identificam o número 8. Daltônicos identificam o número 3. Retirado de [1].	7
2.3	Indivíduos normais identificam o número 2. Daltônicos não conseguem identificar nenhum número. Retirado de [1].	8
2.4	Indivíduos normais não conseguem identificar nenhum número. Daltônicos identificam o número 5. Retirado de [1].	8
2.5	Indivíduos normais identificam o número 26. Daltônicos do tipo Protan só identificam o número 6. Daltônicos do tipo Deutan só identificam o número 2. Retirado de [1].	9
2.6	Cubo com os valores dos eixos normalizados, representando o espaço de cores RGB. Retirado de [2].	12
2.7	Hexagono com os possíveis valores de cada componente do espaço de cores HSV. Retirado de [3].	13
2.8	Exemplo do uso de limiares em um espaço de cores tridimensional. Retirado de [4].	14
3.1	Diagrama de Casos de Uso do Aplicativo. Diagrama elaborado pelo autor.	19
3.2	Diagrama de Sequência para a opção "Realizar teste de Ishihara". Diagrama elaborado pelo autor.	20
3.3	Diagrama de Sequência para a opção "Utilizar a solução de acordo como tipo de daltonismo". Diagrama elaborado pelo autor.	21
3.4	Fluxograma do Aplicativo. Fluxograma elaborado pelo autor.	23
3.5	Primeira Tela de Instruções. Retirado da aplicação móvel executada em um <i>iPad Air</i> com tela de 9,7 polegadas	24
3.6	Segunda Tela de Instruções. Retirado da aplicação móvel executada em um <i>iPad Air</i> com tela de 9,7 polegadas	25
3.7	Terceira Tela de Instruções. Retirado da aplicação móvel executada em um <i>iPad Air</i> com tela de 9,7 polegadas	25

3.8	Tela do teste de Ishihara. Retirado da aplicação móvel executada em um <i>iPad Air</i> com tela de 9,7 polegadas	26
3.9	Tela informando o tipo de daltonismo. Retirado da aplicação móvel executada em um <i>iPad Air</i> com tela de 9,7 polegadas	27
3.10	Tela com mais detalhes do resultado do teste. Retirado da aplicação móvel executada em um <i>iPad Air</i> com tela de 9,7 polegadas	28
3.11	Exemplo de utilização da aplicação criada para auxiliar na detecção dos intervalos de cores. Imagem de outra aplicação móvel criada pelo autor somente para filtrar intervalos de cores no espaço HSV e executada em um <i>iPad Air</i> com tela de 9,7 polegadas	30
3.12	Imagem mostrando o número 6 visto apenas por pessoas sem deficiência. Imagem de outra aplicação móvel criada pelo autor somente para filtrar intervalos de cores no espaço HSV e executada em um <i>iPad Air</i> com tela de 9,7 polegadas.	31
3.13	Imagem mostrando o número 5 visto por pessoas com deficiência. Imagem de outra aplicação móvel criada pelo autor somente para filtrar intervalos de cores no espaço HSV e executada em um <i>iPad Air</i> com tela de 9,7 polegadas.	31
3.14	Imagem mostrando somente a parte em que daltônicos não distinguem. Imagem de outra aplicação móvel criada pelo autor somente para filtrar intervalos de cores no espaço HSV e executada em um <i>iPad Air</i> com tela de 9,7 polegadas.	31
3.15	Pedido para acessar a câmera do <i>tablet</i> . Imagem retirada da aplicação móvel executada em um <i>iPad Air</i> com tela de 9,7 polegadas	32
3.16	Tela contendo instruções de uso. Imagem retirada da aplicação móvel executada em um <i>iPad Air</i> com tela de 9,7 polegadas	33
3.17	Fluxograma da solução utilizando a câmera. Fluxograma elaborado pelo autor.	34
4.1	Exemplo de imagem da identificação de objetos. Retirado da aplicação móvel executada em um <i>iPad Air</i> com tela de 9,7 polegadas	43
4.2	Exemplo de informações adicionais do objeto ao ser selecionado. Retirado da aplicação móvel executada em um <i>iPad Air</i> com tela de 9,7 polegadas	45
5.1	Aplicação detecta a parte problemática da imagem da prancha de número 2, responsável por fazer com que o número 8 seja percebido como número 3 por daltônicos. Retirado da aplicação móvel executada em um <i>iPad Air</i> com tela de 9,7 polegadas	47

5.2	Aplicação detecta a parte problemática da imagem da prancha de número 3, responsável por fazer com que o número 6 seja percebido como número 5 por daltônicos. Retirado da aplicação móvel executada em um <i>iPad Air</i> com tela de 9,7 polegadas	48
5.3	Aplicação detecta a parte problemática da imagem da prancha de número 8, responsável por fazer com que o número 15 seja identificado como o número 17 por indivíduos daltônicos. Retirado da aplicação móvel executada em um <i>iPad Air</i> com tela de 9,7 polegadas	49
5.4	Aplicação detecta o número 2 completo, visto que o indivíduo daltônico não consegue identificar número algum. Retirado da aplicação móvel executada em um <i>iPad Air</i> com tela de 9,7 polegadas	50
5.5	Aplicação detecta o número 73 completo, visto que o indivíduo daltônico não consegue identificar número algum. Retirado da aplicação móvel executada em um <i>iPad Air</i> com tela de 9,7 polegadas	51
5.6	Aplicação detecta o número 7 completo, visto que o indivíduo daltônico não consegue identificar número algum. Retirado da aplicação móvel executada em um <i>iPad Air</i> com tela de 9,7 polegadas	52
5.7	Aplicação detecta o número 6 somente, já que indivíduos do tipo Deutan só enxergam o número 2. Retirado da aplicação móvel executada em um <i>iPad Air</i> com tela de 9,7 polegadas	53
5.8	Aplicação detecta o número 2 somente, já que indivíduos do tipo Protan só enxergam o número 6. Retirado da aplicação móvel executada em um <i>iPad Air</i> com tela de 9,7 polegadas	54

Lista de Tabelas

2.1 Incidência dos principais tipos de daltonismo de forma hereditária	6
2.2 Relação de pranchas do teste de Ishihara	10
3.1 Caso de Uso 1.	19
3.2 Caso de Uso 2	19

Capítulo 1

Introdução

1.1 Contextualização

Desde a Antiguidade pensadores já estudavam e buscavam conhecer mais sobre as cores e sua percepção. Aristóteles (384-322 a.C.) [5] já estudava as cores e analisava que existiam cores principais e que todas as outras eram resultantes de misturas de cores [6]. Posteriormente, Leonardo Da Vinci demonstra, por intermédio de experimentos no seu livro "Tratado da pintura e da paisagem", que a cor branca é a união de todas as outras cores [7]. Uma das maiores contribuições para o estudo das cores ocorreu quando Isaac Newton (1642-1727) utilizou um prisma e conseguiu decompor a luz solar em sete faixas de luzes com cores distintas, provando a decomposição das cores [8] [9]. Desta forma, Newton demonstra que a luz é a fonte de todas as cores [10].

As cores são objetos de estudo desde a Antiguidade, porém, somente em 1794 foi publicado o primeiro estudo relacionado a discromatopsia, principal deficiência visual relacionada as cores, quando John Dalton (1766 - 1844) descreveu por meio deste estudo sua dificuldade para distinguir algumas cores [11]. John Dalton passou a estudar mais sobre o daltonismo e descreveu-o cientificamente em 1798 [12]. Então, o termo daltonismo foi utilizado para discromatopsia por causa dos estudos realizados por John Dalton.

Então, desde 1798 essa deficiência visual já é estudada por mais pesquisadores e científicas e já não é algo distante da população de modo geral. Entretanto, ainda é possível observar no dia a dia um número grande de indivíduos que descobrem de uma maneira tardia que são daltônicas, ou ainda pior, indivíduos que possuem esta deficiência mas que não sabem que possuem seja por falta de informação, por falta de diagnóstico ou até mesmo por desinteresse pessoal.

Com o grande avanço científico na área tecnológica, o uso de tecnologias se tornam cada vez mais utilizadas por pessoas no mundo todo e, uma de suas utilidades, está na área da saúde, que usa as tecnologias como forma de auxílio, seja na prevenção de diagnósticos por

meio de ressonâncias magnéticas ou tomografia computadorizada, seja na gestão da área médica, proporcionando atendimento de melhor qualidade, seja fornecendo informações para o paciente e permitindo o compartilhamento de informações com profissionais da saúde de maneira mais rápida e eficiente por meio da tecnologia móvel [13], a qual permite o uso de aplicativos até mesmo para enviar e receber diagnósticos, e no mapeamento genética, o qual é gerado com o auxílio de tecnologias específicas [14]. Ou seja, os avanços tecnológicos fornecem facilidades tanto na prevenção como no tratamento de doenças.

No âmbito acadêmico já é possível encontrar pesquisas envolvendo aplicativos para dispositivos móveis como *smartphones* e *tablets* que buscam auxiliar no tratamento, prevenção e até diagnóstico de determinadas deficiências e doenças [15] [16].

1.2 Objetivos

Apesar dos avanços tecnológicos na área da saúde, inclusive com a utilização de aplicativos móveis, citados e do maior nível de conhecimento da sociedade de modo geral sobre o daltonismo, o problema que este trabalho tem como objetivo de solucionar é que muitas pessoas ainda não sabem do que se trata o daltonismo e não tem a oportunidade sequer de saber se possuem daltonismo, visto que aproximadamente 10% da população masculina no Brasil possui esta deficiência [17].

O uso de aplicativos móveis para auxiliar no cotidiano de indivíduos com alguma deficiência ou de pessoas com alguma doença tem se mostrado uma ferramenta útil e prática na vida dos pacientes. Aplicando isso no contexto do daltonismo, o objetivo deste trabalho é o desenvolvimento de um aplicativo móvel que possa ser utilizado como uma ferramenta, de fácil acesso, de auxílio no cotidiano de pessoas com daltonismo, permitindo um diagnóstico do daltonismo de maneira prática e fornecendo uma ferramenta que auxilie o daltônico, em tempo real, a identificar elementos do mundo real em que possui dificuldade de distinção de suas cores.

Este objetivo geral pode ser subdividido em dois objetivos mais específicos: implementar um teste, dentro do aplicativo, capaz de diagnosticar o daltonismo do usuário, caso ele possua; utilizar a câmera do dispositivo móvel para identificar todos os objetos do mundo real em que o usuário, após diagnosticado o tipo de daltonismo pelo teste feito no aplicativo, possui dificuldade ou não consiga identificar as cores de forma correta.

Esta subdivisão em objetivos menores e mais específicos foi feita para facilitar o entendimento e a implementação.

Além de buscar ajudar o maior número de indivíduos que possuem daltonismo, este trabalho também teve como motivação uma ferramenta que pudesse auxiliar o cotidiano do autor deste trabalho, que possui daltonismo.

1.2.1 Objetivo Secundário

Além do principal objetivo, este trabalho também visa a divulgação do daltonismo por meio do próprio aplicativo, visto que quanto maior for o número de usuários deste aplicativo, maior será o número de pessoas que, ao menos, terão acesso a informações sobre o daltonismo.

1.3 Distribuição do Manuscrito

O Capítulo 2 apresenta uma fundamentação teórica necessária para o completo entendimento do desenvolvimento e resultado deste trabalho, pois neste capítulo é feita uma introdução sobre os conceitos de daltonismo, sobre o principal método para diagnóstico, visão computacional, sobre a biblioteca utilizada no desenvolvimento do aplicativo e sobre o sistema utilizado no dispositivo móvel.

O Capítulo 3 descreve toda a metodologia proposta para o desenvolvimento do aplicativo móvel criado.

Já no Capítulo 4 são apresentados os detalhes mais técnicos da implementação do aplicativo, detalhes que envolvem trechos de códigos, imagens de telas e funcionamento do mesmo.

No Capítulo 5 é descrito o teste básico, de forma a avaliar como a metodologia proposta pode ser utilizada em uma situação real, em que uma pessoa com discromatopsia interagiria com o sistema proposto como um todo.

O Capítulo 6, que é o último, apresenta a conclusão sobre a utilidade deste trabalho e apresentação de possíveis futuros trabalhos baseados neste.

Capítulo 2

Fundamentação Teórica

Este capítulo descreve definições acerca da Discromatopsia, de conceitos importantes de Visão Computacional e de ferramentas como o OpenCV que auxiliaram no desenvolvimento da solução proposta.

2.1 Daltonismo

Daltonismo, nome mais popular da discromatopsia, é uma deficiência dos cones, localizados no olho, mais precisamente na região central da retina, e que possuem a função de perceber as cores [18]. Um indivíduo normal possui três tipos de cones: um responsável por distinguir o vermelho e suas variantes, outro por distinguir a cor verde e suas variantes e um último por distinguir a cor azul e suas variantes, contemplando assim estas 3 cores principais, as quais fazem com que o indivíduo consiga distinguir todas as cores. É, portanto, uma deficiência que dificulta a percepção das cores, fazendo com que o indivíduo daltônico não consiga ou tenha dificuldade de enxergar determinada(s) tonalidade(s) de cor(es), de acordo com o tipo de daltonismo. É, na grande maioria dos casos, hereditária, isto é, é passada geneticamente de pais para filhos e não possui uma cura.

É estimado que aproximadamente 10% dos homens e 1% das mulheres nascidos no Brasil sejam daltônicos, mesmo não existindo nenhuma pesquisa significativa para levantar esses dados no Brasil [17]. Em outros países, a média fica em aproximadamente 8% e 0,5% da população masculina e feminina, respectivamente, afetada [19].

Porém esses dados apresentados são sobre os tipos de daltonismos mais comuns, o que afetam os cones responsáveis por distinguir o vermelho e o verde, chamados de Protanopia e Deuteranopia, para o caso de deficiência total desses cones, e de Protanomalia e Deuteranomalia, para o caso de deficiência parcial [17].

No decorrer deste trabalho, serão muitas vezes utilizadas as palavras Protan, para representar a Protanopia e Protanomalia, e Deutan, para representar a Deuteranopia e

Deuteranomalia, baseado em uma convenção de se utilizar prefixos gregos para os tipos de daltonismo [20].

2.1.1 Classificações de Daltonismo

Utilizando termos médicos [21], o daltonismo pode ser dividido em monocromatismo, dicromatismo e tricromatismo anômalo. O primeiro, e o mais raro, ocorre em indivíduos que são incapazes de distinguir as cores por possuírem deficiência total em todos os três tipos de cones. Sendo assim, indivíduos com esse daltonismo são caracterizados por somente distinguirem as diferentes tonalidades de cinza e as cores preto e branco.

De acordo com [22], o segundo tipo, o dicromatismo, representa o grupo de daltonismos que geram deficiência total em um dos tipos de cones, resultando em indivíduos que não conseguem distinguir cores na faixa vermelho-verde, os quais possuem o daltonismo do tipo Protanopia, indivíduos que não conseguem distinguir cores no segmento verde-vermelho, possuindo daltonismo do tipo Deuteranopia, e aqueles incapazes de distinguir cores no segmento azul-amarelo, possuindo, portanto, daltonismo do tipo Tritanopia.

O terceiro e último tipo é o tricromatismo anômalo, que é considerado um daltonismo de grau médio ou leve. Afeta algum tipo de cone de maneira parcial, apenas dificultando a distinção e reconhecimento das cores, de acordo com o tipo de cone afetado, podendo então afetar cores na região vermelho-verde, verde-vermelho ou azul-amarelo, resultando, respectivamente, em daltonismos dos tipos Protanomalia, Deuteranomalia ou tritanomia [23].

2.1.2 Incidência dos Tipos

De acordo com os autores Bruni e Cruz [24], os tipos mais comuns encontrados são as formas Protan e Deutan, relacionadas às deficiências de vermelho e verde, enquanto que o tipo mais raro é a tritan.

Quando o daltonismo é herdado, o que ocorre na grande maioria dos casos, a deficiência dos cones ocorre graças a uma deficiência no cromossomo X. Desta forma, como o homem é formado por apenas um cromossomo X enquanto a mulher possui dois, a incidência do daltonismo herdado é maior em indivíduos do sexo masculino do que no sexo feminino, já que nas mulheres são necessário dois cromossomos X afetados para a presença do daltonismo. [25]

A discromatopsia herdada acaba afetando mais homens do que mulheres em daltonismos do tipo Protan e Deutan, seja Protanopia, Protanomalia, Deuteranopia ou Deuteranomalia. A única exceção para incidência iguais em homens e mulheres ocorre no tipo Tritan, sendo Tritanomalia ou Tritanopia, pois esse tipo afeta um gene autossômico

dominante, fazendo com que apenas um gene afetado resulte em um indivíduo afetado, diferentemente dos outros tipos, onde a mulher precisa ter os dois cromossomos X afetados para ser considerada daltônica. [26]

A Tabela 2.1 mostra a relação de incidência entre homens e mulheres nos principais tipos de daltonismo em um levantamento realizado na Inglaterra [27].

Tabela 2.1: Incidência dos principais tipos de daltonismo de forma hereditária

Condição	Homens (%)	Mulheres (%)
Protanopia	1	0,02
Protanomalia	1,5	0,03
Deuteranopia	1	0,01
Deuteranomalia	5	0,40
Tritanopia/Tritanomalia	0,001	0,001

2.2 Teste de Ishihara

Para saber se um indivíduo possui discromatopsia existem alguns métodos, os quais em sua grande maioria são pequenos testes que são capazes de identificar se o paciente possui algum tipo de daltonismo e, até mesmo, determinar qual o tipo de severidade do daltonismo. Entretanto, a eficiência e eficácia varia de teste para teste.

Para a realização deste trabalho foi utilizado, então, o método mais conhecido de avaliação cromática, o teste de Ishihara [28], que foi publicado pela primeira vez em 1917 e até hoje é o mais utilizado e reconhecido teste para esse fim. Possui algumas versões distintas, sendo somente alterada a quantidade de imagens utilizadas. Neste trabalho, foi utilizada a versão com 38 imagens, mas utilizadas somente as primeiras 25 imagens.

O teste de Ishihara possui uma ótima precisão [29] [30], sendo capaz de classificar de maneira muito eficaz entre os tipos Protan e Deutan, tendo melhores resultados para a classificação de Deutans em relação aos Protans.

2.2.1 Funcionamento do Teste

Conforme o apresentado por Ishihara [1], as imagens do teste são chamadas de pranchas, sendo que cada prancha possui uma imagem com vários círculos de diversas cores formando números e linhas. Neste trabalho, foram utilizadas somente pranchas com números.

As pranchas são compostas por círculos coloridos de diversos tamanhos, sendo que em cada prancha as cores agrupadas formam um número. As cores usadas para compor os números são cores em que daltônicos do tipo Protan e Deutan possuem deficiência, fazendo com que estes não consigam identificar corretamente o número, podendo não

perceber número algum, identificar um número diferente, perceber parcialmente o número existente ou até mesmo identificar um número em pranchas que indivíduos considerados normais não identificam número algum.

As pranchas podem ser classificadas em [18]:

- Pranchas de Demonstração - prancha única que possui um número em que indivíduos normais e com qualquer tipo de discromatopsia conseguem identificar. A Figura 2.1 apresenta o único exemplo presente deste tipo de prancha no teste.

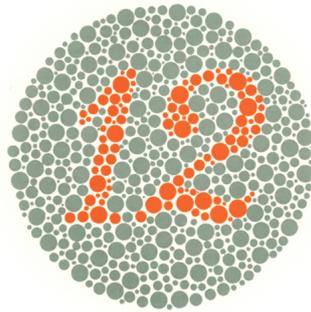


Figura 2.1: Todos conseguem identificar o número 12. Retirado de [1].

- Pranchas de Transformação - pranchas nas quais indivíduos normais identificam um número, enquanto indivíduos com discromatopsia identificam outro número, já que as cores dos círculos exploram justamente a faixa de cores em que Protans e Deutans possuem alguma deficiência. A Figura 2.2 é um exemplo de prancha deste tipo, já que daltônicos identificam outro número quando comparado com indivíduos sem daltonismo.

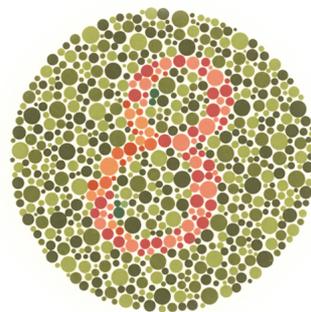


Figura 2.2: Indivíduos normais identificam o número 8. Daltônicos identificam o número 3. Retirado de [1].

- Pranchas de Desaparecimento - somente indivíduos normais conseguem identificar o número na imagem. Daltônicos não conseguem identificar nenhum número. Exemplo apresentado na Figura 2.3 mostra uma prancha em que daltônicos não percebem o número 2.

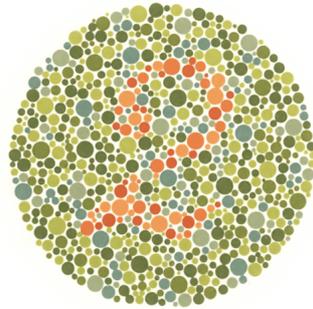


Figura 2.3: Indivíduos normais identificam o número 2. Daltônicos não conseguem identificar nenhum número. Retirado de [1].

- Pranchas Escondidas - somente daltônicos conseguem identificar o número contido na prancha, enquanto um indivíduo normal não consegue identificar número algum na prancha. Neste tipo, os círculos coloridos são agrupados para que somente daltônicos identifiquem um número. Esta classificação, exemplificada na Figura 2.4, é importante para evitar que o paciente tente ludibriar o teste somente afirmando que não identifica número algum em todas as pranchas, já que, necessariamente, em pelo menos alguma das pranchas deste tipo, o daltônico identifica um número.

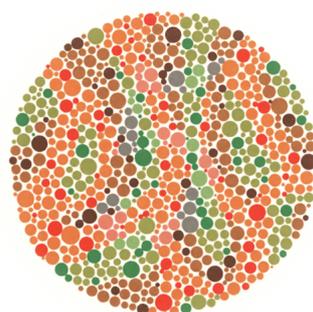


Figura 2.4: Indivíduos normais não conseguem identificar nenhum número. Daltônicos identificam o número 5. Retirado de [1].

- Pranchas de Classificação - em cada prancha deste tipo existem dois números distintos, sendo que daltônicos do tipo Protan conseguem perceber somente um número (e quando o daltonismo não é de grau alto, consegue perceber levemente o outro também), enquanto que daltônicos do tipo Deutan conseguem identificar somente o outro número (da mesma forma dos Protans, Deutans de grau mais leve conseguem perceber de maneira fraca o outro número). Indivíduos normais identificam os dois números. Na Figura 2.5 cada tipo de daltonismo citado acima identifica um numeral do número 26.

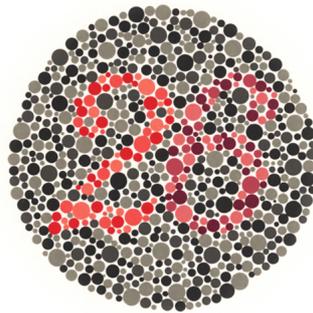


Figura 2.5: Indivíduos normais identificam o número 26. Daltônicos do tipo Protan só identificam o número 6. Daltônicos do tipo Deutan só identificam o número 2. Retirado de [1].

2.2.2 Instruções do Teste

De acordo com o que foi apresentado por Ishihara [1], as instruções do teste afirmam que a iluminação direta do sol ou de luz elétrica pode produzir algumas discrepâncias nos resultados devida a alteração nas sombras das cores nas pranchas.

As pranchas devem ser seguradas na frente do paciente a uma distância de $75cm$ aproximadamente.

O paciente deve então responder se consegue identificar algum número em cada uma das pranchas e, em caso de resposta afirmativa, deve informar qual o número identificado em não mais do que 3 (três) segundos [1].

2.2.3 Análise dos Resultados do Teste

A análise do teste é realizada com base na compilação dos resultados gerados pela execução do teste. A Tabela 2.2 é a referência para a verificação dos resultados e responsável pela geração do diagnóstico relativo ao tipo de daltonismo.

A Tabela 2.2 está presente no livro de Ishihara e é acompanhada de uma detalhada análise do resultado esperado para cada tipo de prancha e o que o resultado representa. A letra X presente na 2.2 representa que a prancha não pode ser lida para aquele tipo de indivíduo, enquanto que um campo em branco representa que a leitura é indefinida para aquela prancha e para aquele tipo específico de indivíduo.

Tabela 2.2: Relação de pranchas do teste de Ishihara

Número da prancha	Pessoa Normal	Pessoa com Deficiências dos tipos Vermelho-Verde				Pessoa monocromata
		Protan		Deutan		
		Forte	Moderado	Forte	Moderado	
1	12					12
2	8					X
3	6					X
4	29					X
5	57					X
6	5					X
7	3					X
8	15					X
9	74					X
10	2					X
11	6					X
12	97					X
13	45					X
14	5					X
15	7					X
16	16					X
17	73					X
18	X					X
19	X					X
20	X					X
21	X					X
		Protan		Deutan		
		Forte	Moderado	Forte	Moderado	
22	26	6	(2)6	2	2(6)	
23	42	2	(4)2	4	4(2)	
24	35	5	(3)5	3	3(5)	
25	96	6	(9)6	9	9(6)	

As pranchas de 1 a 21 determinam se o indivíduo possui percepção das cores normal ou alguma deficiência. Se 17 ou mais pranchas forem lidas de acordo com uma pessoa normal, o indivíduo é diagnosticado como sem daltonismo. Por outro lado, se somente 13 ou menos pranchas forem lidas corretamente, o indivíduo é considerado daltônico. Segundo o autor deste método, é raro encontrar um indivíduo considerado de visão normal que responda de maneira correta de 14 a 16 pranchas [1].

Esse teste possui então 5 tipos distintos de pranchas. A prancha de número 1 é lida por qualquer pessoa, possuindo visão normal ou possuindo alguma deficiência nos cones. As pranchas de número 2 a 9 são aquelas em que pessoas com daltonismo do tipo Deutan ou Protan visualizam um número distinto do que o número visto por pessoas com visão normal. Já as pranchas de número 10 a 17 possuem números que somente indivíduos sem daltonismo conseguem enxergar. Existe ainda as pranchas em que ocorre o contrário, ou seja, somente pessoas com daltonismo do tipo Deutan e Protan conseguem identificar o número presente nas pranchas de número 18 a 21. Por fim, as últimas 4 pranchas, de número 22 a 25, tem o objetivo de diferenciar o tipo de daltonismo. Isso ocorre de acordo com o número observado pelo indivíduo em cada uma dessas pranchas e, de acordo com a resposta do indivíduo, é possível determinar se este possui Protanopia, Protanomalia, Deuteranopia ou Deuteranomalia.

2.3 Visão Computacional

Visão Computacional é definida tradicionalmente como uma disciplina que busca investigar as questões computacionais e algorítmicas associadas à aquisição, ao processamento e à compreensão de imagens [31]. Ainda segundo o apresentado por Trivedi [31], Visão Computacional somada com Neurofisiologia e Psicologia Perceptual são as disciplinas que estudam a visão.

Marr [32] define a visão como sendo o processo que produz, a partir de imagens do mundo externo, uma descrição que é útil ao usuário e que não é repleta de informações irrelevantes.

Sendo assim, a Visão Computacional foi bastante utilizada neste trabalho como ferramenta fundamental para alcançar os objetivos propostos, já que esta forneceu todas as bases teóricas e algorítmicas para o funcionamento correto do aplicativo.

2.3.1 Espaços de Cores

Espaços de cores são padrões criados para poder representar todas as cores, visto que nomear as cores é insuficiente para abranger as cores existentes, sem contar que, de modo geral, grande parte da população associa uma enorme variedade de cores a somente um

único nome [33]. São representações numéricas utilizando sequências ordenadas, isto é, com a ordem importando para representar todas as cores possíveis [34].

RGB

No espaço de cores denominado por RGB, as cores são formadas com base na mistura de três canais, o vermelho(R - Red), verde(Green - G) e o azul(B - Blue). Cada canal pode ter o seu valor variado de 0, representando a ausência deste canal, até 255, que é o valor máximo para um canal [34]. Desta forma, uma cor é representada pela junção dos 3 canais. A cor vermelha, por exemplo, é representada no espaço de cor RGB como $(R = 255, G = 0, B = 0)$. Partindo deste ponto, é possível misturar as cores básicas para formar todas as outras cores.

É muito utilizado no armazenamento e representação de imagens em meio digital [35].

A representação gráfica deste sistema, conforme apresenta a Figura 2.6, é no formato de um cubo, sendo que os três eixos que compõem o cubo são representados pelo R, G e B.



Figura 2.6: Cubo com os valores dos eixos normalizados, representando o espaço de cores RGB. Retirado de [2].

HSV

Outro espaço de cores bastante conhecido e utilizado é o HSV, o qual especifica as cores com valores intuitivos por meio de 3 componentes: Tonalidade (Hue - H) é o que define a cor predominante dentre todas as cores possíveis, com os seus valores indo de 0 até 360° ou de 0 a 100% quando normalizados, já que sua representação geométrica é um hexágono; Saturação (Saturation - S) define a pureza de uma cor, isto é, quanto menor o valor nessa componente, a mais a cor aparecerá com tom de cinza, e seus valores vão de 0 a 100%; e

Valor (Value - V) que determina o brilho da cor, variando de 0 até 100% [34] [36] [37]. A Figura 2.7 ilustra a forma geométrica utilizada para representar o espaço de cores HSV.

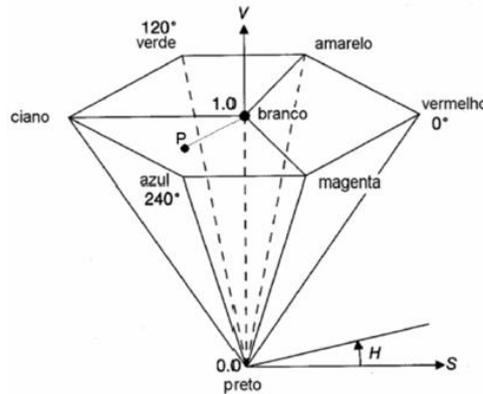


Figura 2.7: Hexágono com os possíveis valores de cada componente do espaço de cores HSV. Retirado de [3].

2.3.2 Segmentação por Cor

De acordo com Ballard & Brown [38], uma imagem segmentada é resultado de um agrupamento de partes de uma imagem qualquer em unidades que sejam homogêneas com respeito a uma ou mais características.

Segundo estes autores, os seres humanos desenvolvem agrupamento visual com base em diversas características, como continuidade, similaridade, proximidade e cores. [38] Na visão computacional, esse agrupamento visual resultam na segmentação, onde são extraídos parâmetros que permitem a associação de regiões da imagem com objetos encontrados na cena [38].

Então, a segmentação por cor acontece por meio de limiares, que é uma técnica que utiliza da representação de uma imagem em um determinado espaço de cores e da similaridade entre os valores dos pixels para ser realizada o agrupamento e, conseqüentemente, a segmentação [4]. Sendo assim, é esperado que se possa determinar os parâmetros de cores para os quais limiares, isto é, os limites superior e inferior, possam processar a segmentação [38]. Para exemplificar a utilização de limiares na segmentação por cores, a Figura 2.8 demonstra que a representação da cor de um determinado pixel de uma imagem deve estar entre os limites, impostos pelos limiares, das três faixas de cores no caso de um espaço de cores tridimensional.

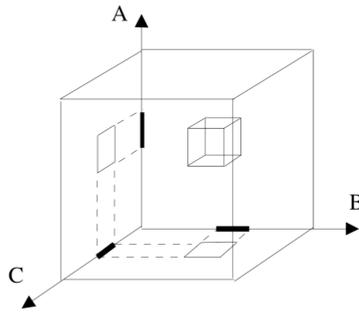


Figura 2.8: Exemplo do uso de limiares em um espaço de cores tridimensional. Retirado de [4].

2.4 OpenCV

OpenCV - Open Source Computer Vision Library [39] é, como o próprio nome já diz, uma biblioteca gratuita de visão computacional e machine learning. Foi criado para prover uma infraestrutura comum para visão computacional e para acelerar o uso da percepção visual de máquinas em produtos comerciais.

Possui mais de 2500 algoritmos otimizados, os quais podem ser usados para: detectar faces; identificar objetos; classificar atividades humanas em vídeos; rastrear movimentos por meio da câmera; agrupar imagens para produzir uma imagem em alta resolução de uma cena específica; rastrear objetos em movimento; encontrar imagens similares a uma imagem em uma base de dados; remover olhos vermelhos em uma fotografia tirada utilizando o flash; seguir o movimento dos olhos e outras várias funcionalidades [40].

Atualmente têm mais de 14 milhões de downloads já realizados [40]. Diversas empresas renomadas utilizam a biblioteca do OpenCV, como a Google, Yahoo, Microsoft, Intel, IBM, Sony, etc. Possui interface em C++, C, Python, Java e MATLAB, possuindo suporte para Windows, Linux e Mac OS.

Foram utilizadas diversas funcionalidades do OpenCV neste trabalho, as quais serão melhor descritas na Metodologia. OpenCV foi uma ferramenta crucial para a realização deste trabalho, permitindo, por exemplo, fazer o processamento de cada frame capturado pela câmera do dispositivo.

Uma das grandes vantagens dessa biblioteca é a vasta documentação encontrada em diversos sites e fóruns de dúvidas, permitindo a rápida solução de problemas encontradas na elaboração do código. Possui também exemplos de aplicações para um mais fácil entendimento sobre cada tipo de funcionalidade que o OpenCV explora.

2.5 Sistema *iOS* & Objective-C

iOS é o sistema operacional móvel da Apple utilizado em todos os seus dispositivos [41]. Foi originalmente criado especificamente para o iPhone, smartphone criado pela empresa, mas depois foi também utilizado em iPads, tablets criados pela mesma empresa [42].

É exclusivamente feito para produtos criados pela Apple [41]. Esse sistema permite a criação de aplicativos, os quais podem ser disponibilizados na loja de aplicativos, chamada de App Store [43]. Para publicar na loja, é preciso antes ter uma conta de desenvolvedor criada e autorizada pela própria Apple [44].

Geralmente, um desenvolvedor iOS precisa conhecer uma das duas linguagens de programação utilizadas na criação de aplicativos para tal sistema, a linguagem Objective-C, que foi utilizada na realização da solução deste trabalho, e Swift, a mais nova linguagem de programação criada pela Apple [45] [46].

A escolha desse Sistema Operacional se deve, primeiramente, à grande utilização de dispositivos móveis no mundo atualmente. Segundo o site especializado em pesquisa Statista, no ano de 2016 aproximadamente 62,9% da população mundial já possuía um smartphone [47], enquanto que, em 2015, pelo menos 1 bilhão de pessoas no mundo todo já eram usuários de tablets [48], o que mostra a grande utilização destes dispositivos atualmente. De acordo com uma pesquisa realizada pela StatCounter Global Stats, em novembro de 2016 foi constatado pela primeira vez que, no mundo todo, as pessoas utilizavam mais smartphones e tablets do que computadores de mesa para acessar a internet, mostrando, mais uma vez, a grande utilização desses dispositivos ao redor do mundo [49].

A linguagem de programação Objective-C foi utilizada devida à sua grande facilidade de uso e por ser extremamente poderosa, contendo uma grande variedade de métodos [45].

Capítulo 3

Metodologia Proposta

Este capítulo descreve a metodologia utilizado para o desenvolvimento da aplicação móvel sugerida para a solução do problema inicial apresentado no Capítulo 1.

3.1 Descrição Inicial da Solução Proposta

A solução proposta então se trata de uma aplicação móvel, mais especificamente para tablets com o sistema *iOS*, sistema criado e utilizado pela Apple em seus dispositivos [41]. Foi escolhido desenvolver para *tablets* já que esta categoria de dispositivo móvel possui, além de uma tela maior que facilita o uso da aplicação, a possibilidade de simulação de um óculos inteligente, como por exemplo o *Glass* [50], por meio da utilização da câmera traseira do dispositivo, facilitando uma futura migração para dispositivos óculos inteligentes.

Foi então proposta a ideia de elaborar um aplicativo que auxiliasse o dia a dia de um indivíduo com daltonismo, criando uma ferramenta que auxilie o usuário a identificar todos os objetos no mundo real que possui uma ou mais cores em que este usuário possui dificuldade ou não consegue perceber, de acordo com o tipo de daltonismo diagnosticado.

De forma mais detalhada, o aplicativo então é composto de duas etapas: A primeira etapa é composta pelo teste de Ishihara [1] para que o usuário tenha, de modo acessível, um meio de saber se possui daltonismo e, ainda mais, qual o tipo de daltonismo que possui se for o caso. Mas mais importante ainda, o aplicativo armazenará a informação do teste, isto é, o resultado produzido pelo usuário, para utilizar na solução com a câmera, que é a segunda etapa. Ainda na primeira etapa, será apresentado para o usuário o resultado do teste de Ishihara, mostrando o tipo de daltonismo e informando os resultados esperados para cada uma das pranchas utilizadas.

Já na segunda etapa, que é a principal, o sistema utiliza, como já foi dito, o resultado do teste realizado na primeira etapa. A segunda etapa utiliza a câmera do dispositivo

para capturar as imagens do mundo externo e, então, o sistema aplica métodos de visão computacional para filtrar, nas imagens capturadas, qualquer objeto que possua uma ou mais cores nas quais o usuário não consiga, totalmente ou parcialmente, distingui-la(s). Por fim, é destacado para o usuário todos estes objetos identificados, colocando um contorno branco em volta de cada um deles, para que o usuário consiga perceber em tempo real tudo aquilo que está enxergando de uma maneira alterada do normal. O usuário poderá ainda interagir com as imagens capturadas, selecionando determinado contorno produzido pelo aplicativo para ver, de forma mais detalhada, a composição de cores do objeto identificado.

3.1.1 Motivação

A motivação deste trabalho de graduação surgiu com uma motivação pessoal do autor deste trabalho, o qual possui daltonismo e sempre almejou uma solução simples e prática que pudesse auxiliá-lo em sua rotina, permitindo que dificuldades enfrentadas durante alguma atividade do dia pudesse ser mais facilmente vencida com o auxílio de uma ferramenta dessa natureza.

Além disso, visto o tamanho da parcela da população que possui discromatopsia e a porcentagem de pessoas que utiliza tablets hoje em dia, é muito provável que essa solução possa ser útil para mais pessoas que sofrem com essa deficiência visual.

Então, essa solução poderá ser de grande utilidade para a sociedade, auxiliando os indivíduos, desde descobrirem se possuem ou não algum tipo de daltonismo, até a já fornecer uma ferramenta que ajude estes que forem diagnosticados com alguma discromatopsia no seu dia a dia.

3.1.2 Requisitos do Aplicativo

De acordo com o Institute of Electrical and Electronics Engineers (IEEE), requisito é uma condição ou capacidade que um usuário necessita para resolver um problema ou alcançar um objetivo. Uma condição ou capacidade que deve ser satisfeita por um sistema para satisfazer um contrato ou um padrão [51].

Também pode ser definido como o que estabelece o que o sistema deve fazer, além de definir as restrições a respeito de sua operação e implementação [52].

Sommerville [52] ainda classifica os requisitos de software como funcionais e não funcionais. Desta forma, foram definidos os requisitos, tanto funcionais como não-funcionais, listados abaixo:

Requisitos Funcionais:

- Usuário deve receber instruções básicas iniciais sobre o daltonismo e sobre o funcionamento do aplicativo
- Usuário deve realizar o teste de Ishihara
- Usuário deve obter o resultado do teste logo após a realização
- Aplicativo deve fornecer resultado detalhado do teste realizado
- Aplicativo deve utilizar a câmera do dispositivo
- Aplicativo deve mostrar em tempo real os objetos que contenham uma ou mais cores diagnosticadas como problemáticas de acordo com o resultado do teste
- Usuário deve interagir com a aplicação, podendo visualizar mais detalhadamente a composição das cores de um objeto identificado
- Usuário deve ser capaz de refazer o teste de Ishihara
- Usuário deve ser capaz de congelar a imagem capturada em vídeo para mais fácil interação

Requisitos Não-Funcionais:

- Software deve ser feito para sistema iOS
- Software deve ser feito especificamente para tablet
- Quantidade de frames por segundo não deve ser muito baixa para não interferir na experiência do usuário
- Aplicativo deverá ajustar os filtros aplicados na câmera de acordo com o resultado do teste

3.1.3 Modelagem do Sistema

Utilizando a UML - Unified Modeling Language, foram criados o diagrama de casos de uso e dois diagramas de sequência, detalhando os casos de uso apresentados. Para a criação dos diagramas, foi utilizado o Software StarUML, disponível para a plataforma Mac OS.

Diagrama de Casos de Uso

Esse tipo de diagrama possui o objetivo de ilustrar um possível cenário do sistema, apresentando as funcionalidades do sistema na perspectiva de um ator, que no caso aqui apresentado, será o usuário do aplicativo.

O aplicativo possui dois possíveis cenários, representados na Figura 3.1:

O primeiro cenário é apresentado de forma mais detalhada pela Tabela 3.1.

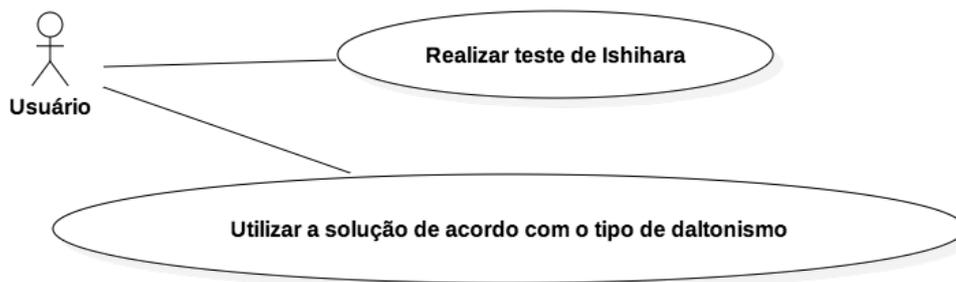


Figura 3.1: Diagrama de Casos de Uso do Aplicativo. Diagrama elaborado pelo autor.

Tabela 3.1: Caso de Uso 1.

Realizar o Teste de Ishihara	
Ator	Usuário do aplicativo
Pré-Condições	Aplicativo instalado no dispositivo do usuário
Pós-Condições	Resultado do teste apresentado para o usuário informando se possui daltonismo ou não e, se possuir, informa o tipo de daltonismo e armazena essa informação
Descrição	Usuário deseja realizar o teste de Ishihara para descobrir se possui daltonismo e qual o tipo de daltonismo. Aplicativo então inicia o teste, mostrando pranchas do teste de Ishihara, fornecendo um teclado para o usuário digitar a resposta e passando para a próxima prancha até o término do teste. Quando o teste é finalizado, é mostrado para o usuário o resultado do teste, mostrando se possui ou não daltonismo.

O segundo cenário é apresentado na Tabela 3.2.

Tabela 3.2: Caso de Uso 2

Utilizar a solução de acordo como tipo de daltonismo	
Ator	Usuário do aplicativo
Pré-Condições	Aplicativo instalado no dispositivo do usuário e teste de Ishihara já realizado
Pós-Condições	Sem pós-condições
Descrição	Usuário já realizou o teste de Ishihara para descobrir se possui daltonismo e qual o tipo de daltonismo. Aplicativo então utiliza o resultado do teste para aplicar o filtro apropriado para o tipo específico de daltonismo do usuário. Usuário lê as instruções e começa a utilizar a solução, permitindo que o usuário possa interagir em tempo real com a imagem capturada em vídeo pela câmera do dispositivo móvel, aonde são detalhados os objetos em que o usuário possui dificuldade de identificar alguma cor.

A seguir, são apresentados os diagramas de sequência que detalham como ocorre o fluxo de eventos em cada um dos casos de uso.

Diagramas de Sequência

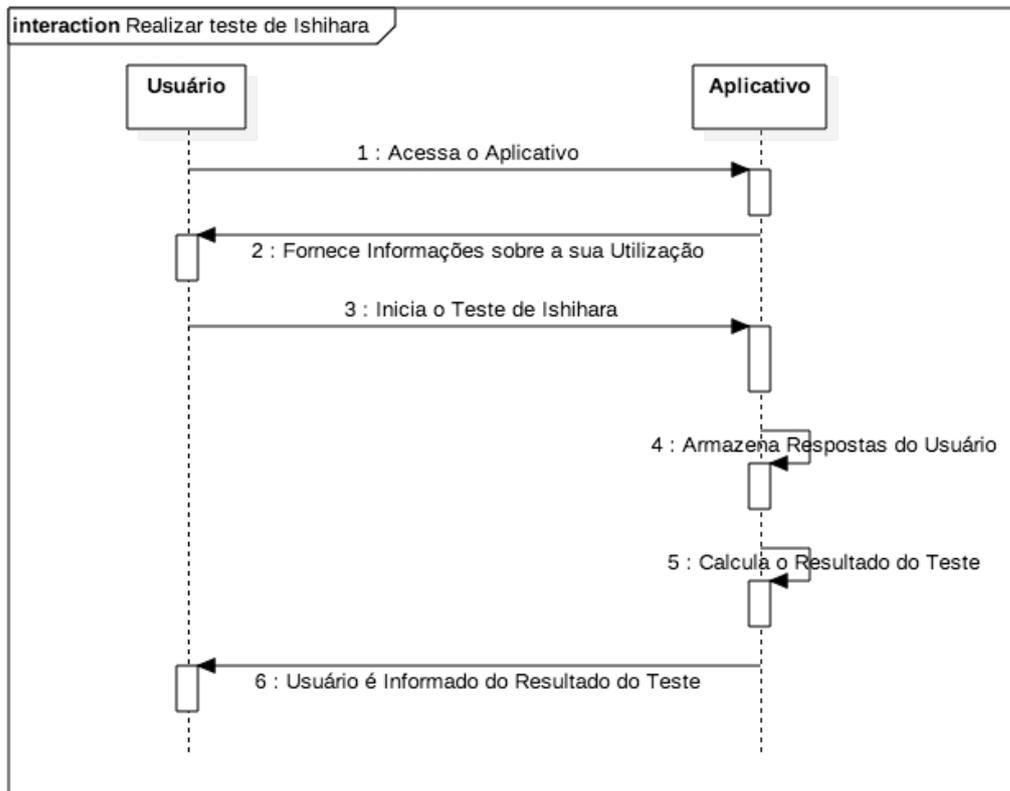


Figura 3.2: Diagrama de Sequência para a opção "Realizar teste de Ishihara". Diagrama elaborado pelo autor.

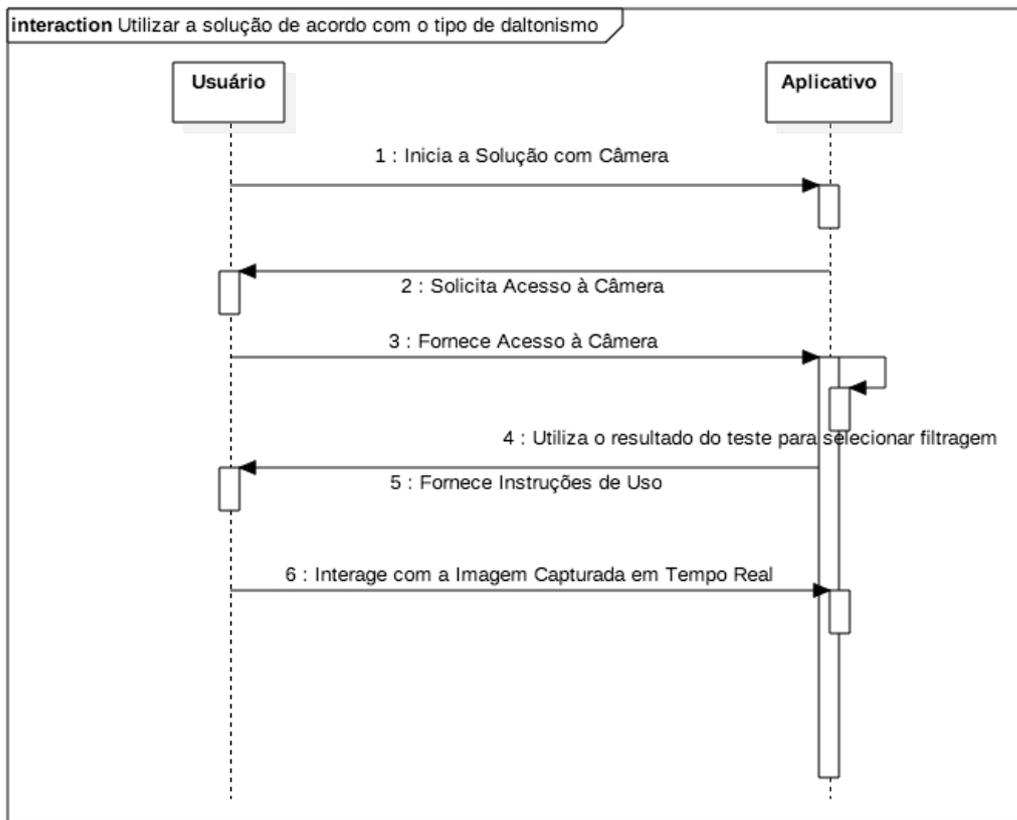


Figura 3.3: Diagrama de Sequência para a opção "Utilizar a solução de acordo como tipo de daltonismo". Diagrama elaborado pelo autor.

3.1.4 Funcionalidades

Com a fase de levantamento de requisitos concluída e com os diagramas já desenhados, as funcionalidades principais do aplicativo foram então definidas para a determinação do escopo do projeto, visando focar somente no mais importante para o bom funcionamento do aplicativo, além de cumprir os objetivos iniciais deste trabalho.

- Realização do teste de Ishihara - O teste de Ishihara é realizado assim que o usuário inicia a utilização do aplicativo pela primeira vez. Depois de realizado, o usuário pode realizar o teste quantas vezes quiser.
- Identificação de objetos que o usuário possui dificuldade - O Aplicativo, de acordo com o tipo de daltonismo, identificará para o usuário, por meio da câmera do dispositivo móvel, os objetos em que o usuário pode não enxergar da maneira correta devido à sua deficiência visual.

- Interação com objetos identificados - O usuário poderá obter mais informações da composição de cores de cada objeto identificado ao selecionar um contorno, que representa um objeto identificado pelo sistema.
- Pausar imagem - O usuário poderá também pausar a imagem que está sendo capturada pela câmera, para maior facilidade de interação com objetos identificados pelo sistema.

3.2 Desenvolvimento da Aplicação Móvel

O aplicativo foi desenvolvido por meio da ferramenta *Xcode*, que é um Ambiente de Desenvolvimento Integrado - IDE - criado pela *Apple*. Essa IDE é especializada na criação de aplicativos para a plataforma *iOS*, sendo esse o principal motivo para a sua escolha no processo de criação do aplicativo deste trabalho. O *Xcode* já vem com todas as ferramentas necessárias para a criação de um aplicativo móvel para o sistema *iOS*. O download dessa IDE pode ser feito pela loja de aplicativos para computadores com sistema *Mac OS* de forma gratuita [53].

Além do *Xcode*, foi necessário a utilização de um *iPad Air* com tela de 9,7 polegadas para testar a aplicação no próprio dispositivo. O *Xcode* fornece um simulador de dispositivos da *Apple*, para que o desenvolvedor possa testar seus aplicativos antes de publicá-los na loja de aplicativos, retirando a necessidade então de o desenvolvedor possuir um dispositivo próprio para testes. Porém, no caso deste aplicativo específico, como é totalmente necessária a utilização da câmera do dispositivo para o objetivo proposto pelo aplicativo, não foi possível utilizar a ferramenta de simulação de dispositivos, sendo então utilizado o *iPad Air* com tela de 9,7 polegadas do autor para testes.

Para o auxílio no processo de desenvolvimento do aplicativo, foi elaborado um fluxograma para esclarecer cada passo que o aplicativo deverá possuir.

3.2.1 Fluxograma

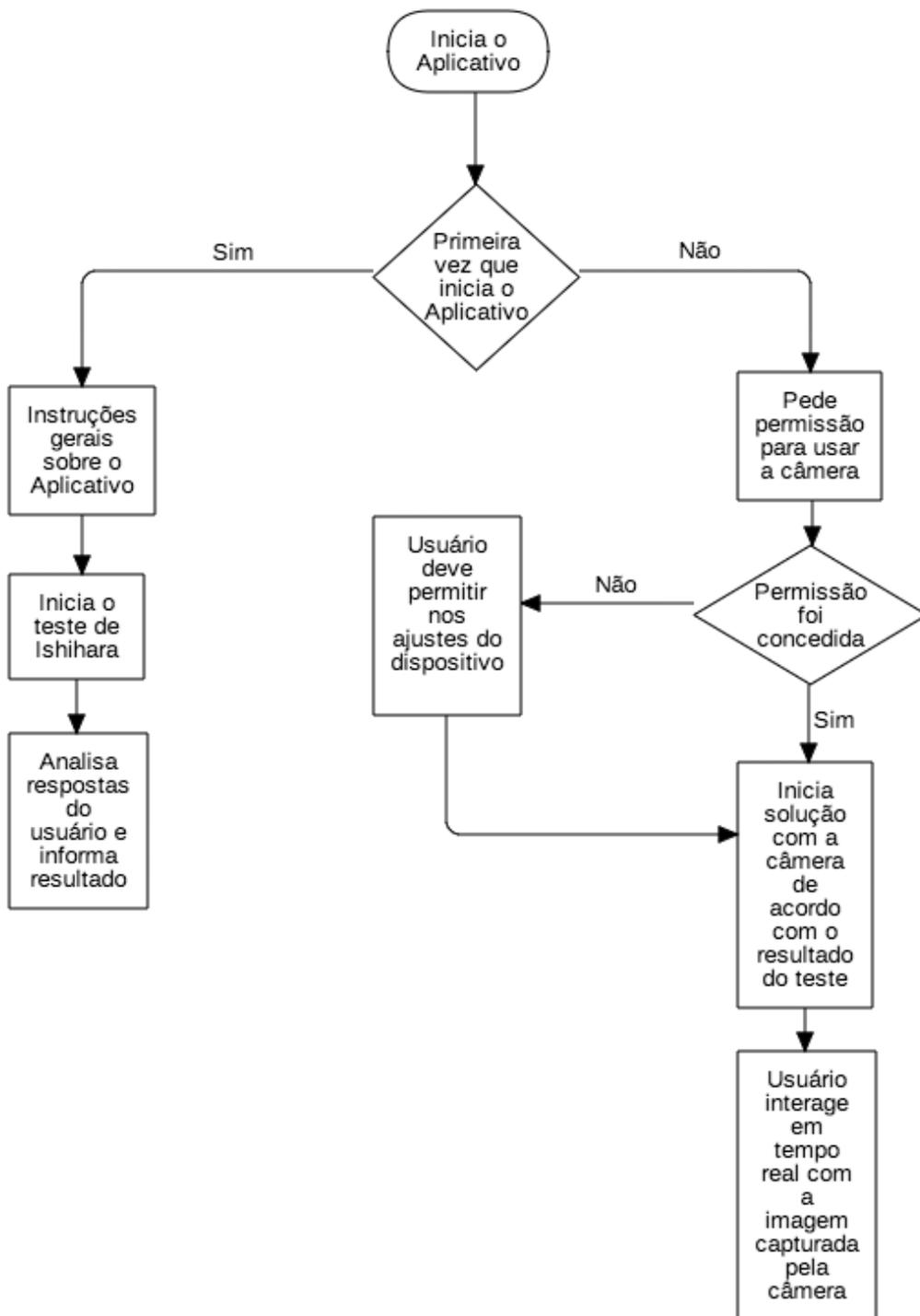


Figura 3.4: Fluxograma do Aplicativo. Fluxograma elaborado pelo autor.

3.2.2 Criação da Apresentação Inicial

Com base no fluxograma anteriormente apresentado, foi iniciado o processo de criação das telas do aplicativo. Não foi necessária a utilização de nenhuma ferramenta para criar protótipos das telas, visto que o *Xcode* possui uma ferramenta de criação de interfaces, chamada de *Interface Builder*, a qual fornece um ambiente completo para a criação de telas sem ser necessária a utilização de nenhum código.

A *Interface Builder* torna simples a criação de telas, pois possui diversos elementos utilizados em aplicativos no sistema *iOS* que podem simplesmente ser arrastados para a tela no local desejado. É possível fazer isso com botões, janelas, textos, imagens, tabelas e diversos outros elementos já fornecidos no *Xcode* [53]. Sendo assim, a criação inicial das telas, utilizadas como protótipos, foram elaboradas por meio dessa ferramenta, permitindo inclusive testar as telas criadas no dispositivo utilizado, facilitando e tornando mais eficiente a fase de prototipagem.

O aplicativo se inicia, quando nunca aberto anteriormente, em três telas que possuem breves informações sobre o daltonismo, o teste de Ishihara e a respeito da solução com câmera utilizada após o teste. Essas telas possuem o objetivo de fazer uma breve apresentação sobre do que se trata o aplicativo, explicando os tipos de daltonismo, o método de Ishihara e o objetivo da solução utilizando a câmera.



Figura 3.5: Primeira Tela de Instruções. Retirado da aplicação móvel executada em um *iPad Air* com tela de 9,7 polegadas

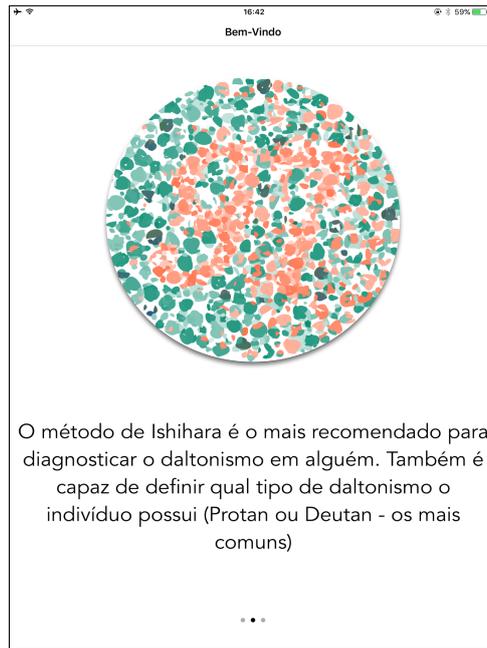


Figura 3.6: Segunda Tela de Instruções. Retirado da aplicação móvel executada em um *iPad Air* com tela de 9,7 polegadas

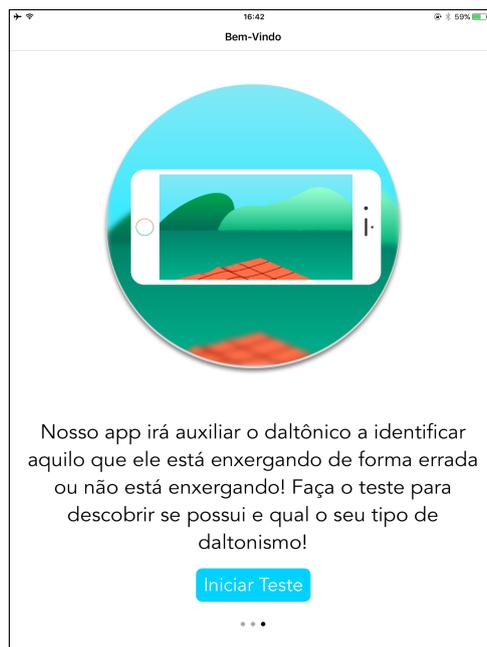


Figura 3.7: Terceira Tela de Instruções. Retirado da aplicação móvel executada em um *iPad Air* com tela de 9,7 polegadas

3.2.3 Teste de Ishihara no Aplicativo

Como visto na Figura 3.7, existe um botão que leva diretamente para a tela onde o teste de Ishihara é realizado dentro do aplicativo.

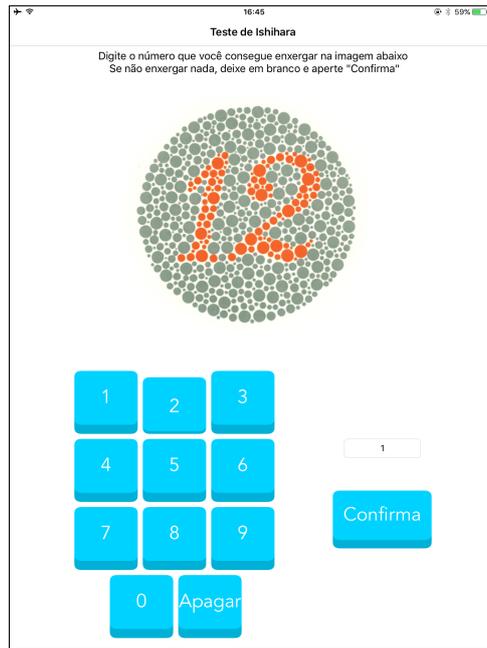


Figura 3.8: Tela do teste de Ishihara. Retirado da aplicação móvel executada em um *iPad Air* com tela de 9,7 polegadas

Foram utilizadas 25 pranchas do teste. No aplicativo, o usuário irá olhar para a imagem na parte superior da tela e, rapidamente, tentará identificar o número contido na imagem. Então, deverá selecionar os botões que formam o número identificado e, após o número aparecer na caixa de texto do lado direito, deverá selecionar o botão "Confirmar". O botão "Confirmar" muda a imagem atual para a próxima imagem do teste, apagando o que foi escrito na caixa de texto.

Quando na última prancha do teste, ao usuário selecionar o botão para confirmar sua resposta, o aplicativo mudará de tela, passando para a tela de resultados do teste.

Nessa tela, em vez de fornecer quatro ou cinco opções para o usuário selecionar qual seria a resposta correta, foi preferível utilizar o teclado numérico para que o usuário não seja influenciado pelas opções fornecidas, respondendo algo em que o usuário não está vendo, o que alteraria o resultado do teste e todo o objetivo da aplicação.

3.2.4 Telas com o Resultado do Teste

Ao finalizar o teste, o aplicativo passa para a tela de Resultados, a qual irá informar se o usuário possui ou não daltonismo e, se possuir, qual o tipo de daltonismo. Para gerar essa informação, foram utilizadas as instruções contidas no livro de Ishihara [1], o qual descreve de maneira detalhada as respostas esperadas para cada tipo de daltonismo e para pessoas sem daltonismo.

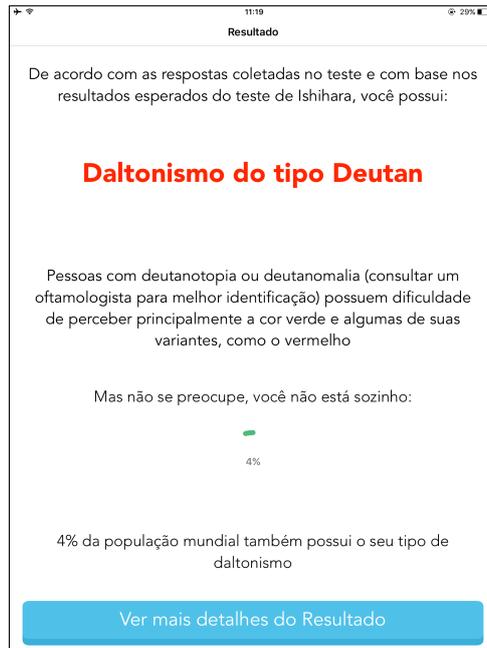


Figura 3.9: Tela informando o tipo de daltonismo. Retirado da aplicação móvel executada em um *iPad Air* com tela de 9,7 polegadas

A Figura 3.9 apresenta para o usuário se este possui daltonismo e qual o tipo de daltonismo, apresentando uma breve descrição da principal dificuldade do tipo especificado de daltonismo, além de apresentar um gráfico circular mostrando a porcentagem da população que possui o mesmo tipo de daltonismo apresentado.

Ao final da Figura 3.9, o botão, quando selecionado, passa para a próxima tela de resultados, a qual possui informações mais detalhadas das respostas do usuário e das respostas esperadas no teste de acordo com Ishihara. São apresentados também 3 gráficos na parte superior mostrando a porcentagem de respostas iguais as esperadas para pessoas normais, pessoas com daltonismo do tipo Protan e pessoas com daltonismo do tipo Deutan.

O usuário consegue percorrer na horizontal os gráficos na parte superior da Figura 3.10 e percorrer verticalmente os detalhes das respostas esperadas para cada prancha pois foi utilizada uma *ScrollView*, que é uma *View*, isto é, uma porção da janela atual, que no caso é a tela do aplicativo, que pode ser rolada para qualquer direção de acordo com o tamanho do conteúdo presente na *ScrollView*. Essa ferramenta é uma das inúmeras utilizadas para o desenvolvimento deste aplicativo para *iOS*, as quais são gratuitamente fornecidas e já instaladas e configuradas no *Xcode*.

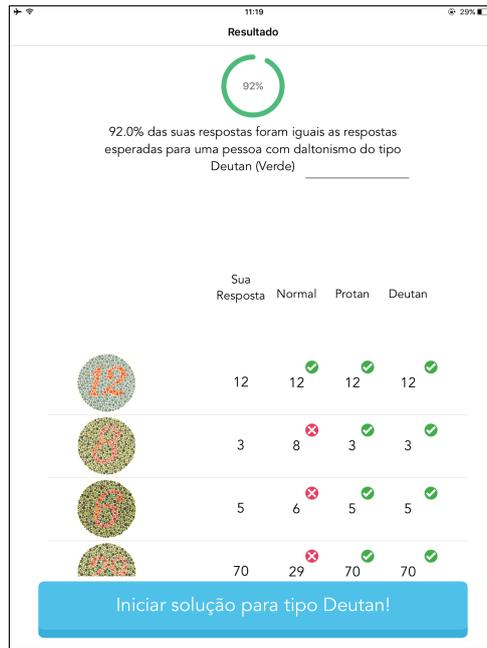


Figura 3.10: Tela com mais detalhes do resultado do teste. Retirado da aplicação móvel executada em um *iPad Air* com tela de 9,7 polegadas

3.2.5 Importância da Determinação do Tipo de Discromatopsia

Foram apresentados para o usuário os resultados do teste que foi realizado e o usuário então já possui uma informação valiosa, sabendo se possui daltonismo e qual o seu tipo de daltonismo. Porém, essa etapa de realização do teste de Ishihara e verificação do resultado do teste é uma parte crucial para o bom funcionamento da próxima parte do aplicativo, que é a utilização da câmera como ferramenta de auxílio para os daltônicos. Isto porque é a determinação do tipo de Discromatopsia que o usuário possui que irá determinar os filtros adequados para aquele tipo de usuário, permitindo uma melhor experiência de uso do aplicativo.

O objetivo do aplicativo é proporcionar para um indivíduo daltônico um aplicativo que pudesse auxiliá-lo, no seu cotidiano, na identificação de objetos em que o indivíduo possui dificuldade de identificar as cores de maneira correta. E para isso, é extremamente necessária a correta determinação do tipo de daltonismo, já que o aplicativo busca aplicar os filtros mais apropriados de acordo com as respostas do usuário no teste de Ishihara.

Na subseção 3.2.9 será apresentada de maneira detalhada como o resultado do teste é utilizado nos filtros aplicados na câmera do dispositivo, ficando então mais claro a importância da determinação do tipo de daltonismo do usuário.

3.2.6 Estratégica de Identificação de Cores Problemáticas

Cores problemáticas são as que o daltônico possui dificuldade de distingui-las, isto é, são as cores em que o indivíduo com daltonismo não consegue identificar, ou acaba por confundi-las com outras cores próximas, justamente por não possuir os cones responsáveis pela percepção das mesmas.

Portanto, para determinar os intervalos de cores que um indivíduo com alguma discromatopsia possui deficiência, foram utilizadas as imagens de cada uma das pranchas do teste de Ishihara.

A ideia para determinação das cores foi baseada na tabela de resultados presentes no livro do método de Ishihara [1], o qual possui as respostas esperadas para cada tipo de indivíduo, seja para um indivíduo sem discromatopsia, ou com discromatopsia do tipo Deutan ou Protan e até mesmo para indivíduos com daltonismo mais grave, distinguindo apenas tons de cinzas no mundo real.

Com base nessas informações, a metodologia adotada foi esta:

- Para uma determinada prancha, foi primeiramente analisado e anotado os números esperados para serem observados por pessoas com daltonismo e sem daltonismo, visto que, com exceção das últimas quatro pranchas utilizadas neste trabalho, em todas as outras as respostas esperadas por daltônicos de tipos diferentes são as mesmas.
- Com o conhecimento das respostas esperadas, foi criada, pelo autor, uma aplicação específica que utiliza a biblioteca do *OpenCV* somente com o objetivo de auxiliar, de uma maneira rápida e eficaz, a detecção das partes de cada imagem do teste que são responsáveis pela alteração da resposta esperada de uma pessoa com e sem daltonismo. Esse processo de detecção ocorreu da seguinte maneira:
 - Primeiramente a imagem foi convertida do sistema de cores BGR para HSV, sendo que o HSV é composto por Hue (Tonalidade), Saturation (Saturação) e Value (Valor). De acordo com a documentação do *OpenCV*, esse sistema de cores possui valores que vão de 0 até 180 para a Tonalidade (H), e de 0 até 255 para a Saturação (S) e o Valor (V) [54].
 - Foram criados mecanismos que facilitam a seleção de um valor em um determinado intervalo. Por exemplo, para selecionar o valor 100 em um intervalo que vai de 0 até 255, foi criado então um *slider*, que permite a mais fácil e ágil escolha do valor desejado. Os *sliders* criados estão na parte inferior da Figura 3.11.

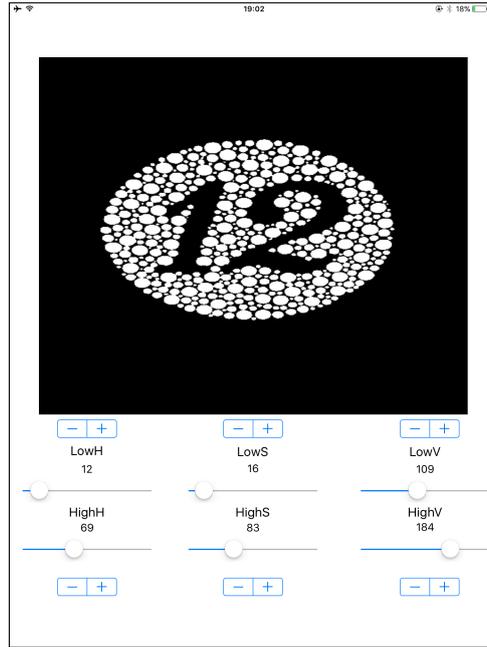


Figura 3.11: Exemplo de utilização da aplicação criada para auxiliar na detecção dos intervalos de cores. Imagem de outra aplicação móvel criada pelo autor somente para filtrar intervalos de cores no espaço HSV e executada em um *iPad Air* com tela de 9,7 polegadas

- Na parte inferior da aplicação então, com o apoio dos *sliders* criados, é determinado o valor mínimo e máximo para cada um dos componentes do sistema de cores HSV, mostrando em tempo real na imagem localizada no centro da tela somente as partes da imagem que possuem cores dentro destes intervalos.
- Então, foi detectado para cada imagem o menor intervalo possível de cada componente que mostrasse exclusivamente a parte da imagem que alterava o resultado de pessoas com daltonismo e sem daltonismo. Um exemplo de uma prancha permite um melhor entendimento a respeito disso. Na prancha de número 3, de acordo com a Tabela 2.2, um indivíduo normal identifica o número 6 enquanto indivíduos com daltonismo identificam o número 5. Sendo assim, foi utilizado esta aplicação de apoio para identificar o menor intervalo possível que tivesse unicamente partes da imagem em que, quando não perceptíveis, podem fazer com que o número 6 não seja percebido, e sim o número 5.
- Por fim, foram registrados todos os intervalos de cada componente das imagens para todas as pranchas utilizadas no teste de Ishihara no aplicativo. A identificação destes intervalos é o que permite que a solução utilizando a câmera seja calibrada especificamente para o determinado tipo de indivíduo, já que, de acordo com as

respostas do teste, é possível determinar quais intervalos de cores o indivíduo possui dificuldade de distinguir e perceber cores.

- Os resultados registrados são então utilizados na aplicação de acordo com o tipo de discromatopsia identificado no teste. Sua utilização é descrita no capítulo 4.

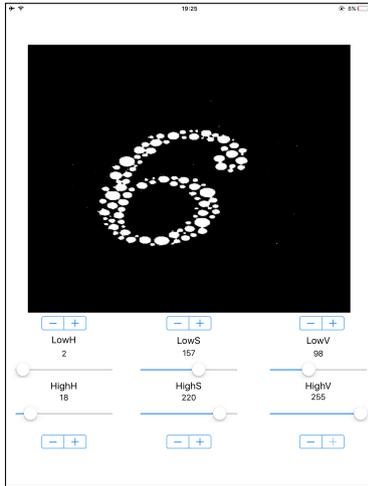


Figura 3.12: Imagem mostrando o número 6 visto apenas por pessoas sem deficiência. Imagem de outra aplicação móvel criada pelo autor somente para filtrar intervalos de cores no espaço HSV e executada em um *iPad Air* com tela de 9,7 polegadas.

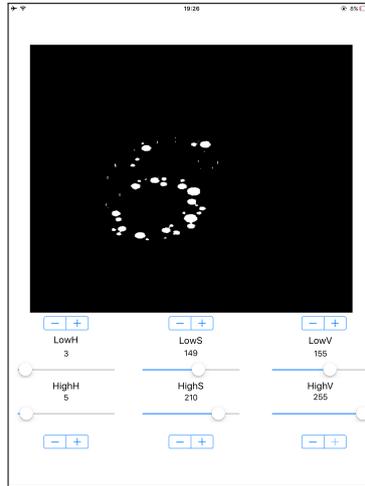


Figura 3.13: Imagem mostrando o número 5 visto por pessoas com deficiência. Imagem de outra aplicação móvel criada pelo autor somente para filtrar intervalos de cores no espaço HSV e executada em um *iPad Air* com tela de 9,7 polegadas.

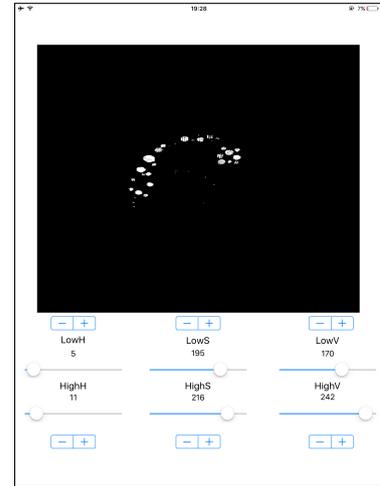


Figura 3.14: Imagem mostrando somente a parte em que daltônicos não distinguem. Imagem de outra aplicação móvel criada pelo autor somente para filtrar intervalos de cores no espaço HSV e executada em um *iPad Air* com tela de 9,7 polegadas.

Esta metodologia inovadora foi desenvolvida pelo autor e foi essencial para desenvolver a funcionalidade do aplicativo de detectar os objetos e elementos do mundo real em que os usuários com daltonismo possuem dificuldade de identificar as suas cores.

3.2.7 Permissão de Acesso e Instruções de Uso

Após a realização do teste de Ishihara e da apresentação do resultado do teste informando o tipo de daltonismo, o usuário, ao apertar o botão localizado na parte inferior da tela Figura 3.10, é levado a próxima tela que contém a ferramenta de auxílio ao tipo especificado de daltonismo no teste.

Antes da solução proposta iniciar, é necessária autorização do usuário para que o dispositivo possa utilizar a câmera do mesmo, como apresentado na Figura 3.15:

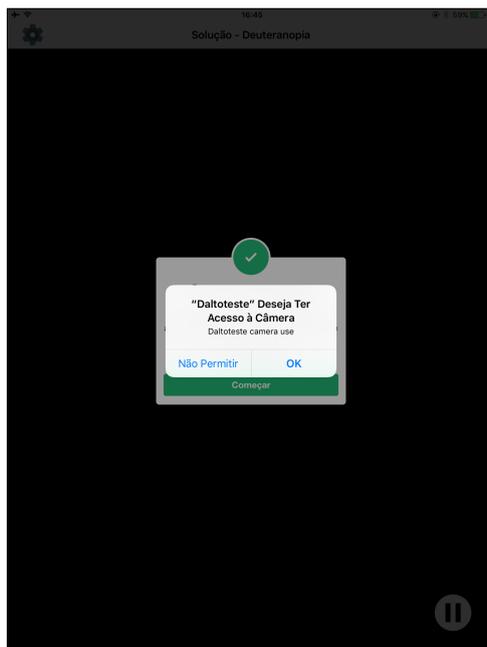


Figura 3.15: Pedido para acessar a câmera da *tablet*. Imagem retirada da aplicação móvel executada em um *iPad Air* com tela de 9,7 polegadas

Ao permitir o acesso, o aplicativo apresenta para o usuário uma pequena janela, como pode ser visto na Figura 3.16, com informações sobre a ferramenta que irá se iniciar, com o intuito de fornecer instruções de uso do aplicativo, além de explicar brevemente o que a ferramenta irá fazer. Essas informações, mesmo que breves, são essenciais para uma boa usabilidade do aplicativo, capacitando indivíduos sem conhecimento algum do aplicativo a entenderem a sua utilização, bem como informando a respeito do objetivo do aplicativo.

Então, a câmera do dispositivo é iniciada para que a ferramenta de auxílio aos daltônicos possa funcionar e ser útil. A seguir, é explicado os detalhes da implementação desta ferramenta, que contou basicamente com o auxílio da biblioteca *OpenCV* e das ferramentas do *Objective-C*, uma das linguagens de desenvolvimento de aplicativos para o sistema *iOS*.

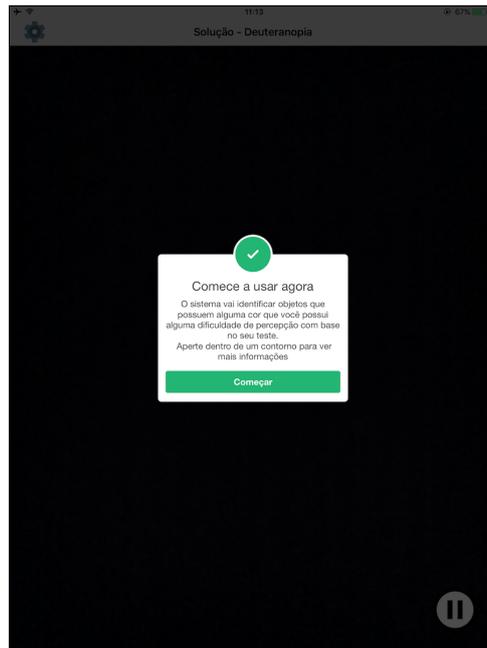


Figura 3.16: Tela contendo instruções de uso. Imagem retirada da aplicação móvel executada em um *iPad Air* com tela de 9,7 polegadas

3.2.8 Utilização do *OpenCV*

Como já informado, a biblioteca *OpenCV* foi a principal ferramenta de apoio para a elaboração desta solução que utiliza a câmera. Foi necessário importar a biblioteca para o *Xcode* para que todas as suas funcionalidades pudessem ser acessadas e utilizadas na aplicação. Após o download da biblioteca específica para plataforma *iOS*, que é fornecida em seu sítio ¹, e de sua devida instalação na máquina na qual foi desenvolvido o aplicativo, foi então configurado no *Xcode* para ser utilizado nesta aplicação. Os detalhes da instalação e configuração não serão apresentados pois não fazem parte do escopo deste trabalho.

Após a devida instalação, basta alterar uma classe criada em *Objective-C* para *Objective-C++* para que a importação de qualquer pacote do *OpenCV* baixado possa ser acessado no código, já que essa biblioteca feita na linguagem de programação *C++*.

A importação dos pacotes do *OpenCV* em uma classe utilizada no aplicativo é feita da seguinte maneira:

```
#include <opencv2/opencv.hpp>
#include <opencv2/videoio/cap_ios.h>
```

Durante a explicação da implementação da solução utilizando a câmera, será explicitado o uso de cada um dos pacotes utilizado na aplicação desenvolvida.

¹<https://opencv.org/releases.html>

3.2.9 Solução Utilizando a Câmera

Para auxiliar no entendimento de como a solução que utiliza a câmera para auxiliar a vida de uma pessoa com daltonismo foi criada, foi elaborado um fluxograma das etapas que a aplicação percorre para cumprir seu objetivo.

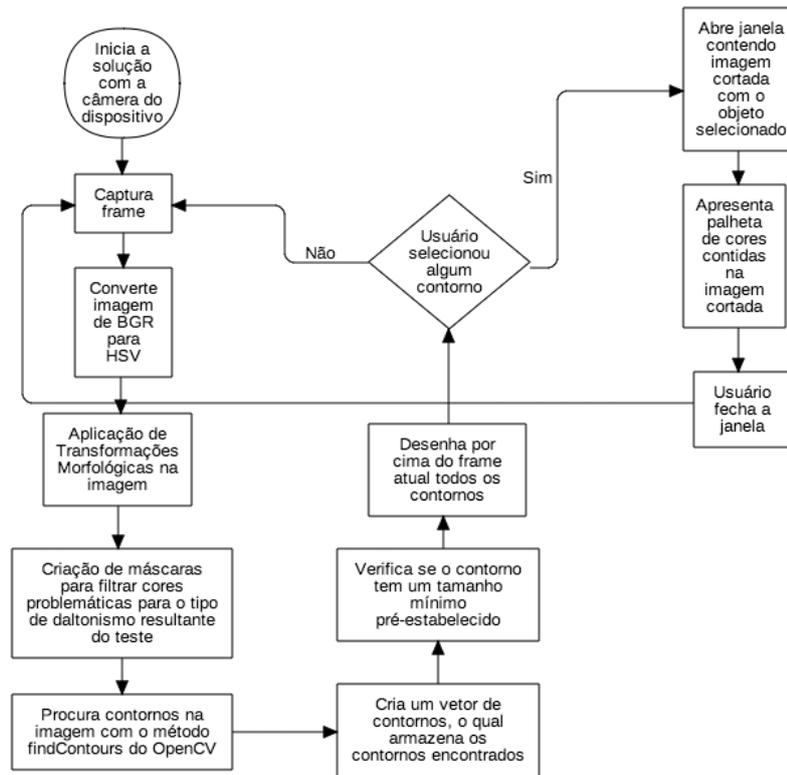


Figura 3.17: Fluxograma da solução utilizando a câmera. Fluxograma elaborado pelo autor.

Foi destinado um capítulo exclusivo para apresentação, em detalhes, de cada etapa apresentada no fluxograma acima.

Capítulo 4

Implementação da Solução

Este capítulo tem o objetivo de descrever cada passo feito na implementação da ferramenta de auxílio para pessoas com daltonismo. A implementação foi toda elaborada com base no fluxograma da imagem Figura 3.17 apresentado na seção 3.2.9 da metodologia.

4.1 Configuração da Câmera

No projeto criado para o desenvolvimento do aplicativo, foi criada uma classe denominada *CameraViewController* para elaboração da solução utilizando a câmera.

Na aplicação desenvolvida, foi incluído o módulo:

```
#include <opencv2/videoio/cap_ios.h>
```

O módulo foi incluído para que fosse utilizada a *CvVideoCamera*, que é uma classe que utiliza a câmera do dispositivo e que contém métodos da biblioteca *OpenCV*, o que facilita a integração entre a câmera do dispositivo utilizado e os métodos da biblioteca.

Após esse passo, foi necessário configurar a câmera para que a imagem capturada fosse apresentada em uma janela específica, que no caso da programação para *iOS* é a *UIImageView*, além de definir se a captura das imagens será feita na vertical ou horizontal, qual o tamanho da imagem capturada, quantos frames por segundo a câmera deverá processar e, por fim, iniciar a câmera por meio do método:

```
self.videoCamera = [[CvVideoCamera alloc] initWithParentView:_imageView];  
...  
[self.videoCamera start];
```

Outra importante configuração a ser feita é ligar o atributo *delegate* da instância da classe *cvVideoCamera* criada à classe atual, no caso a *CameraViewController*. Isso faz com a *CameraViewController* utilize os métodos obrigatórios da classe *cvVideoCamera* por

meio do seu protocolo, que é o que diz quais métodos da classe devem ser implementadas na classe instanciada.

```
self.videoCamera.delegate = self;
```

A linha de código então faz exatamente o que foi dito acima, associa o protocolo da classe *cvVideoCamera* à classe *CameraViewController*, representada na linha de código pela palavra *self*.

4.2 Processamento de Imagem

A partir do momento em que a câmera é iniciada, o dispositivo começa a capturar os frames. Cada frame é armazenado na estrutura *cv::Mat*, que nada mais é do que uma classe que armazena uma imagem em forma de matriz, já que uma imagem para um computador é uma sequência de bits. Essa classe, quando instanciada em forma de um objeto, armazena a imagem em uma matriz, facilitando a sua manipulação.

```
-(void)processImage:(Mat&)image;
```

O método acima, o qual é executado graças ao protocolo da classe *cvVideoCamera*, começa a ser executado ao iniciar a câmera, recebendo como parâmetro cada um dos frames capturados pela câmera do dispositivo.

Este método é executado automaticamente enquanto a câmera estiver funcionando.

Após capturar o frame, inicia-se então o processamento desta imagem, para que sejam aplicados os filtros certos como o objetivo de identificar somente objetos na imagem que possuem cores dentro do intervalo detectado como problemático.

4.2.1 Mudança do Espaço de Cores

Seguindo o fluxograma mostrado na Figura 3.17, após a captura do frame, a imagem precisa ser convertida do espaço de cores BGR para o HSV. Como informado no sítio oficial ¹ que contém a documentação do *OpenCV* para o sistema *iOS*, o frame capturado está no espaço de cores BGRA. Para a conversão para o sistema de cores desejado e adequado para a aplicação, a biblioteca do *OpenCV* fornece um método de conversão de cores, o qual possui como parâmetros a imagem de entrada, a imagem de saída e um parâmetro informando a conversão desejada.

¹https://docs.opencv.org/2.4/doc/tutorials/ios/video_processing/video_processing.html

```
cvtColor(image, image, CV_BGRA2BGR);  
cvtColor(image, image, CV_BGR2HSV);
```

É necessário então converter antes do espaço de cores *BGRA* para *BGR*, o que na verdade somente remove a camada de transparência da imagem, para somente depois ser possível a conversão para *HSV*, pois a biblioteca não fornece um parâmetro do tipo *CV_BGRA2HSV*.

Foi utilizado o padrão HSV para que a imagem fosse mais facilmente segmentada em preto e branco, onde, na imagem resultante após aplicação de filtros de intervalos de cores, o que está com a cor branca é o que passou pelo filtro.

4.2.2 Transformações Morfológicas

Com a imagem já convertida para HSV, são feitas transformações morfológicas na imagem, isto é, operações na imagem que processa a imagem baseada em formas geométricas [55]. Possuem como objetivo a eliminação de ruído na imagem, isolar ou agrupar elementos dentro da imagem e a de tornar a imagem com menos imperfeições. As duas principais operações são a *Erosão* e a *Dilatação*.

Erosão

Resumidamente, a Erosão diminui as partículas presentes em uma imagem, aumenta os buracos e separa elementos conectados de acordo com o tamanho do elemento estruturante passado como parâmetro nesta função [56]. A biblioteca *OpenCV* fornece estruturas prontas para serem utilizadas como elemento estruturante, definindo a forma e o tamanho do elemento que irão atuar na erosão da imagem. Na aplicação foi utilizada a forma geométrica elíptica de tamanho 3.

Dilatação

Já a Dilatação gera o efeito quase contrário, já que as partículas são aumentadas em vez de serem diminuídas, os buracos são preenchidos e os elementos próximos em uma imagem são conectados [56]. Foi utilizado, da mesma forma da função de Erosão, a forma geométrica elíptica de tamanho 3.

Abertura

Quando utilizados em sequência, isto é, a função de Erosão é utilizada imediatamente antes da função de Dilatação, é feito um processo denominado Abertura. Este processo é capaz então de eliminar ruídos, ou seja, elimina elementos considerados pequenos na

imagem, além de ser capaz de suavizar o contorno e de separar elementos próximos que eram conectados na imagem original [56].

```
//Opening
erode(img, img, getStructuringElement(MORPH_ELLIPSE, cv::Size(3,3)));
dilate(img, img, getStructuringElement(MORPH_ELLIPSE, cv::Size(3,3)));
```

Fechamento

Após o processo de Abertura, é comum utilizar um processo que irá produzir efeitos contrários ao efeito do primeiro processo, mas sem eliminar os efeitos principais gerados e desejados pela Abertura, isto é, a eliminação de ruído e suavização do contorno [56].

Então, o processo de Fechamento, que nada mais é do que a utilização da função de Dilatação seguido imediatamente da função de Erosão, permite o fechamento dos buracos abertos e conexão dos elementos desconexos próximos.

```
//Closing
dilate(img, img, getStructuringElement(MORPH_ELLIPSE, cv::Size(3,3)));
erode(img, img, getStructuringElement(MORPH_ELLIPSE, cv::Size(3,3)));
```

É importante reforçar que o objetivo principal destas transformações morfológicas utilizadas é a de eliminar possíveis ruídos na imagem, permitindo que o processamento da imagem trabalhe somente com o conteúdo desejado.

4.2.3 Criação de Máscaras de Intervalos de Cores

Este é um dos pontos cruciais para que o objetivo do aplicativo possa ser cumprido, pois é neste passo em que ocorre a filtragem de elementos com cores dentro as detectadas como problemáticas.

A biblioteca *OpenCV* possui uma função capaz de filtrar todos os pixels de uma determinada imagem de entrada nos quais a sua cor esteja dentro de um intervalo determinado de cores que são passados como parâmetros nesta função.

A função descrita acima é a *inRange()*, que possui como parâmetros:

- Matriz de entrada, armazenada na estrutura *cv::Mat*, que contém a imagem original
- Vetor de três elementos contendo o limiar inferior de cada uma das componentes do espaço de cor utilizado, ou seja, um vetor em que:
 - Primeiro elemento representa o valor mínimo para a componente Hue - Tonalidade

- Segundo elemento representa o valor mínimo para a componente Saturation - Saturação
- Terceiro elemento representa o valor mínimo para a componente Value - Valor

Os valores das componentes podem ir de 0 até 255, com exceção da componente Hue, que vai de 0 até somente 180.

- Outro vetor com a mesma estrutura do anterior, mas este contém os valores máximos para cada componente, representando então o limiar superior do valor que representa a cor do pixel.
- Matriz de saída, também armazenada em uma estrutura do tipo *cv::Mat*, a qual irá conter a imagem filtrada. Esta imagem será criada da seguinte maneira: será verificado na matriz original todos os pixels que estão entre o limiar inferior e superior que foram passados como parâmetros na função *inRange*. Nas posições dos pixels que estiverem entre os limiares, serão então desenhados pixels brancos na matriz de saída, alterando o valor do pixel para 255 em todas as suas componentes, o que o faz ter a cor branca. Desta forma, a imagem de saída terá, na cor branca, todos os pixels da imagem de entrada que estão contidos dentro o intervalo de cores informados como limiares [57].

Então, no aplicativo, a função foi utilizada para cada intervalo de valores descoberto de acordo com a metodologia proposta na subseção 3.2.6, isto é, para cada uma das pranchas do teste de Ishihara.

```
inRange(img, Scalar(5,144,201), Scalar(16,207,249), mask2);
```

Para cada intervalo de cor registrado, a matriz resultante foi armazenada em uma máscara diferente.

Como já explicado na subseção 2.2.1, existem diferentes tipos de pranchas no teste. Desta forma, para pranchas pertencentes ao mesmo tipo, foram encontrados intervalos de cores iguais na maioria dos casos. Por isso, foram criadas 9 máscaras diferentes, as quais estão declaradas e usadas no trecho de código a seguir:

```
Mat mask2, mask3To5, mask6To9, mask10To13, mask14And17, mask15And16,
mask22To25, mask22To25Deutan1, mask22To25Deutan2, mask_combined;
```

```
inRange(img, Scalar(5,144,201), Scalar(16,207,249), mask2);
inRange(img, Scalar(7,159,179), Scalar(15,199,239), mask3To5);
inRange(img, Scalar(7,148,214), Scalar(14,190,255), mask3To5);
```

```

inRange(img, Scalar(25,117,85), Scalar(32,158,183), mask6To9);
inRange(img, Scalar(8,145,200), Scalar(18,191,250), mask10To13);
inRange(img, Scalar(32,99,75), Scalar(52,126,204), mask14And17);
inRange(img, Scalar(42,134,49), Scalar(70,188,156), mask15And16);
if (colorBlindType == 1) { //Deutan
    inRange(img, Scalar(174,147,70), Scalar(180,185,121), mask22To25Deutan1);
    inRange(img, Scalar(0,129,193), Scalar(3,169,252), mask22To25Deutan2);
    Mat mask_deutan = mask22To25Deutan1 | mask22To25Deutan2;
} else { //Protan
    inRange(img, Scalar(0,142,193), Scalar(5,167,255), mask22To25);
}

```

Cada máscara então irá possuir uma imagem em preto e branco, sendo que tudo que estiver com a cor branca será aquilo que estava na imagem original e contido dentro dos limiares inferiores e superiores, ou seja, tudo que estiver em branco representará todos os elementos da imagem que possuem alguma cor identificada como problemática para aquele tipo de daltonismo.

Foi criada uma variável, chamada de *colorBlindType*, que informa se o resultado do teste apresentou que o usuário possui ou daltonismo do tipo Deutan ou daltonismo do tipo Protan. Foi necessário criar máscaras diferentes para estes casos, já que as pranchas de número 22 a 25 foram feitas para a distinção entre Protans e Deutans.

Para juntas as diferentes máscaras criadas em uma estrutura única, foi feita uma operação lógica de *OR*, ou seja, como todos as máscaras possuem tamanho iguais e, logicamente, matrizes de tamanho iguais, o resultado desta operação lógica entre duas máscaras irá gerar uma terceira máscara contendo a junção de todos os pixels brancos das duas máscaras. Desta forma, foi feita a operação com todas as máscaras, criando uma nova máscara que é a junção de todas as outras:

```

//Deutan
mask_combined = mask2 | mask3To5 | mask6To9 | mask10To13
| mask14And17 | mask15And16 | mask_deutan;
...
//Protan
mask_combined = mask2 | mask3To5 | mask6To9 | mask10To13
| mask14And17 | mask15And16 | mask22To25;

```

4.2.4 Encontrar e Desenhar Contornos

Após a criação da máscara que contém todos os pixels da imagem original que estão dentro de algum intervalo de cores em que o indivíduo possui dificuldade de distinção, e seguindo o fluxograma apresentado na Figura 3.17, foram criados os contornos em volta dos objetos identificados na máscara. Para a detecção dos contornos, a biblioteca *OpenCV* disponibiliza a função *findContours()*, que é capaz de detectar todos os contornos em uma determinada imagem de entrada e armazenar as coordenadas de cada contorno em um vetor.

De forma sucinta, a função possui como parâmetros uma matriz de entrada que deve conter uma imagem binária, isto é, uma imagem contendo apenas 1s e 0s, representando as cores branca e preta, deve ter como parâmetro também um vetor de saída, o qual será composto por vetores com as coordenadas de cada contorno identificado na imagem de entrada, um outro vetor de saída que não será utilizado neste trabalho mas que armazena informações sobre a topologia da imagem. Além desses parâmetros, existe ainda um parâmetro indicando o modo da obtenção dos contornos, podendo o modo ser do tipo:

- *CV_RETR_EXTERNAL*: a função armazena somente os contornos externos. Se existir um contorno dentro de outro, será armazenado no vetor de saída somente as coordenadas do contorno mais externo. [58]
- *CV_RETR_LIST*: todos os contornos são armazenados, já que os contornos identificados não são hierarquizados. [58]
- *CV_RETR_CCOMP*: hierarquiza os contornos em dois níveis, permitindo então armazenar contornos externos e internos. [58]
- *CV_RETR_TREE*: hierarquiza os contornos em diversos níveis, e é mais utilizado para detectar quais contornos estão dentro de outros. [58]

Para este trabalho, foi utilizado o modo *CV_RETR_LIST*, pois o objetivo do projeto é identificar todos os objetos e elementos da imagem capturada pela câmera em que o indivíduo não consegue ou possuem dificuldade de distinguir sua cor.

Outro parâmetro importante é o que informa o método de detecção dos contornos, que podem ser, principalmente:

- *CV_CHAIN_APPROX_NONE*: armazena todos os contornos identificados sem nenhuma modificação. [58]
- *CV_CHAIN_APPROX_SIMPLE*: compressa horizontalmente, verticalmente ou diagonalmente todos os contornos identificados para depois armazenar os contornos. [58]

Um último parâmetro opcional informa a margem de pontos que cada coordenada pode ser deslocada para encontrar os contornos.

Desta forma, a função *findContours* foi utilizada da seguinte maneira:

```
vector<vector<cv::Point>> contours_combined;  
vector<Vec4i> hierarchy;  
  
findContours(mask_combined, contours, hierarchy,  
            CV_RETR_LIST, CV_CHAIN_APPROX_SIMPLE, cv::Point(0, 0));
```

Após a identificação de todos os contornos presente na imagem de entrada, são removidos os contornos considerados de tamanho muito reduzidos percorrendo todo o vetor de contornos e verificando o seu tamanho, removendo-o ou não do vetor dependendo do seu tamanho.

Então, com os contornos identificados e filtrados pelo tamanho, é utilizada a função *drawContours* que tem como objetivo, desenhar contornos em cima de uma imagem de entrada. Desta forma, foi utilizada como parâmetro de entrada o vetor de contornos criado e como imagem a ser desenhada o frame inicial capturado pela câmera. Desta forma, a imagem original capturada pela câmera somente terá adicionado a ela os contornos dos elementos com cores identificadas como problemáticas. É possível definir a cor dos contornos que serão desenhados, e no caso deste trabalho, foi escolhida a cor branca por ser facilmente identificada e para que indivíduos com discromatopsia não tenham dificuldade de distingui-la, já que não possuem dificuldades com a cor branca.

4.3 Interação com o Usuário

Concluída a etapa de processamento da imagem, onde a imagem passa pelo processo de remoção de ruídos, de filtragem dos pixels que continham cores dentro de algum dos intervalos detectados como problemáticos de acordo com as imagens do teste de Ishihara, de identificação e desenho dos contornos dos objetos identificados, o usuário passa a interagir com o aplicativo.

É apresentada para o usuário, em tempo real, a imagem com os contornos desenhados já de acordo com os frames capturados pela câmera do dispositivo. O usuário já é capaz então de detectar todos os objetos em que possui alguma dificuldade de distinguir alguma cor presente. A partir deste momento, o aplicativo já serve de auxílio para uma pessoa com algum tipo de discromatopsia, pois o indivíduo consegue identificar objetos ao seu redor, em tempo real, nos quais não consegue determinar a cor correta destes devido a

sua deficiência cromática. A Figura 4.1 apresenta uma imagem da aplicação funcionando em um ambiente cotidiano e detectando objetos com cores consideradas problemáticas.

A quantidade de objetos detectados pode variar de acordo com a iluminação do ambiente capturado pela câmera, pela quantidade de objetos presente na cena e, principalmente, pela predominância de cores na cena capturada. A Figura 4.1 mostra como a iluminação, seja natural ou artificial, influencia diretamente na cor percebida pela câmera do dispositivo, fazendo com que um objeto tenha somente uma parte identificada como problemática de acordo com a iluminação.

Esta é uma das principais limitações do aplicativo, já que a iluminação é um fator que não pode ser controlado pelo dispositivo e que influencia diretamente na cor resultante.



Figura 4.1: Exemplo de imagem da identificação de objetos. Retirado da aplicação móvel executada em um *iPad Air* com tela de 9,7 polegadas

O usuário conta ainda com mais uma ferramenta de auxílio para o seu cotidiano. Foi desenvolvida uma funcionalidade no aplicativo capaz de mostrar as 8 principais cores que compõem um determinado objeto identificado como problemático. Isto foi feito porque, geralmente, a cor de um objeto na verdade é a composição de várias cores que, juntas, formam a cor vista pelos olhos humanos. Desta forma, foi desenvolvida esta funcionalidade para que o usuário pudesse ter conhecimento sobre as 8 cores com maior porcentagem de composição da cor observada. Isto permite ao usuário identificar, por exemplo, quais as 8 principais cores que compõem uma determinada tonalidade de verde detectada pelo aplicativo como uma cor problemática para aquele usuário.

Para mostrar as 8 principais cores da imagem, foi utilizada a técnica de *K-Means Clustering*, a qual, basicamente, agrupa os pixels em 8 blocos distintos de acordo com a proximidade de cor e identifica qual a cor que melhor representa cada um dos blocos realizando diversas iterações a fim de identificar a menor distância entre as cores dos pixels de cada bloco [59].

Para exibir essa informação, conforme apresentada na Figura 4.2, o usuário simplesmente deve pressionar a tela do dispositivo dentro de algum contorno identificado e mostrado.

Como a captura de frames é alta por segundo, foi implementado um botão no canto inferior direito que permite congelar a imagem capturada pela câmera, facilitando a seleção de algum contorno desejado.

Ao selecionar um contorno, será exibida uma janela, onde no centro será mostrado a imagem original com foco no objeto selecionado e, abaixo da imagem, serão apresentados 8 retângulos coloridos com informações abaixo deles. Cada um dos 8 retângulos representa uma das 8 principais cores contidas dentro do contorno selecionado. Abaixo de cada retângulo, é mostrada a porcentagem que existe dessa cor dentro do contorno e seu nome em inglês. Os nomes foram obtidos no sítio² especializados em nomear as cores de acordo com o seu código em hexadecimal.

O usuário consegue fechar a janela aberta tocando em qualquer outra localidade da tela diferente da janela aberta. Ao fechar a janela, é possível selecionar outro contorno para ver informações das cores deste outro.

Se o usuário decidir por pausar a captura de imagem para facilitar a escolha de algum objeto, para reiniciar a captura do vídeo, é necessário tocar no botão inferior direito.

²<http://www.color-hex.com/color-names.html>

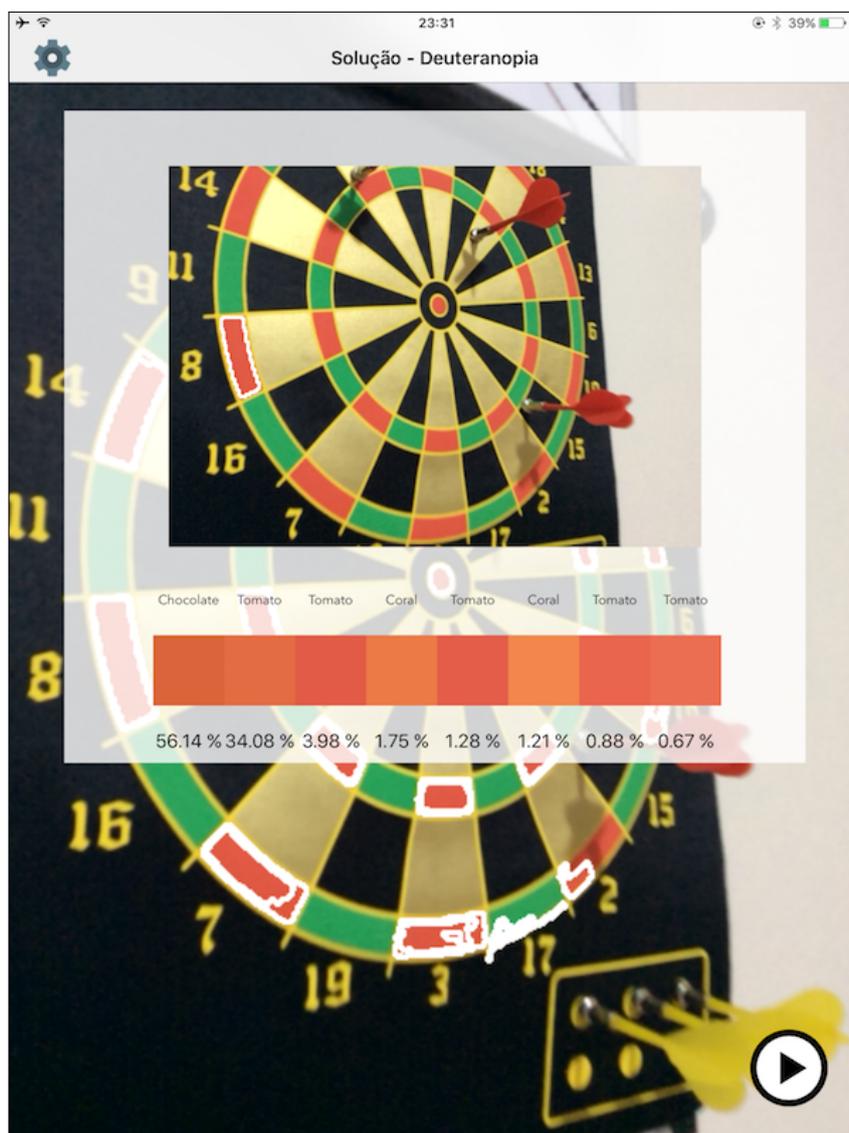


Figura 4.2: Exemplo de informações adicionais do objeto ao ser selecionado. Retirado da aplicação móvel executada em um *iPad Air* com tela de 9,7 polegadas

Capítulo 5

Resultados

Uma das possíveis formas para obter resultados que comprovem a eficácia da aplicação, isto é, cumprir seu objetivo inicial de identificar tudo aquilo que o indivíduo daltônico possui dificuldade de identificar e visualizar no mundo real, foi com a utilização das pranchas do teste de Ishihara e utilização da aplicação desenvolvida de forma a avaliar o que seria visível para um usuário que possui a discromatopsia. Os resultados são detalhados a seguir.

5.1 Obtenção dos Resultados

Para obter os resultados apresentados a seguir, primeiramente todas as imagens do teste de Ishihara foram abertas em um monitor. Em seguida, o aplicativo já instalado no dispositivo móvel foi aberto e colocado em execução, passando então a capturar as imagens do mundo externo por meio de sua câmera. Importante ressaltar que o teste de Ishihara foi feito no aplicativo de acordo com as respostas do autor e o daltonismo foi identificado como do tipo Deuteranopia. Por fim, a aplicação detectou, em cada imagem do teste, todos os elementos considerados problemáticos para um indivíduo com daltonismo do tipo Deutan, sendo que cada elemento tem seu contorno desenhado na cor branca.

É importante frisar a diferença das pranchas que compõem o teste de Ishihara, existindo pranchas de diversos tipos que são compostas por combinações de cores distintas, as quais foram detalhadas na Seção 2.2.1. Isto significa que, para cada tipo de prancha, parâmetros diferentes foram estabelecidos para filtrar somente os elementos desejados, ou seja, os elementos problemáticos para cada tipo de daltonismo. Os filtros estabelecem quais os valores, mínimos e máximos para cada componente do espaço de cores HSV, capazes de filtrar somente os elementos desejados. Os valores dos filtros são armazenados em máscaras, conforme explicado na Seção 4.2.3.

Em alguns casos, mesmo com pranchas distintas porém com combinações de cores próximas, foi necessário criar filtros diferentes para estas pranchas, visto que um filtro capaz de detectar a parte problemática de uma prancha não foi capaz de detectar a parte problemática da outra prancha, mesmo ambas sendo de cores muito próximas, já que uma variação mínima de cores impossibilita que um filtro identifique o número ou parte problemática da prancha de forma correta.

5.2 Filtro para Prancha de Número 2

A prancha de número 2, de acordo com a Tabela 2.2, possui uma imagem na qual indivíduos normais identificam o número 8 enquanto indivíduos com discromatopsia identificam o número 3. O aplicativo, então, quando aplicado somente o filtro para a prancha de número 2, conforme mostrado na Figura 5.1, informa ao usuário a parte do número 8 que, quando não identificada, se transforma no número 3.

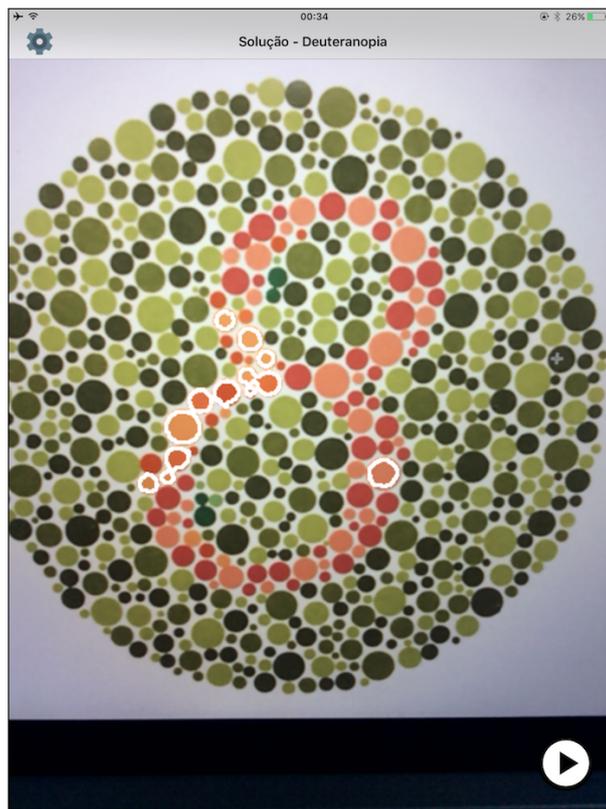


Figura 5.1: Aplicação detecta a parte problemática da imagem da prancha de número 2, responsável por fazer com que o número 8 seja percebido como número 3 por daltônicos. Retirado da aplicação móvel executada em um *iPad Air* com tela de 9,7 polegadas

5.3 Filtro para Pranchas de Números 3 a 5

Das pranchas de número 3 até a prancha de número 5, o mesmo intervalo de cores foi utilizado para identificar a parte problemática da imagem, isto é, a parte em que faz com que o daltônico enxergue um número distinto quando comparado com o número enxergado por uma pessoa normal. Na prancha utilizada na Figura 5.2 como exemplo, a parte detectada pelo aplicativo é justamente o que cria a diferença do número 6 para o número 5, dificultando assim a percepção do número 5 para os indivíduos daltônicos.



Figura 5.2: Aplicação detecta a parte problemática da imagem da prancha de número 3, responsável por fazer com que o número 6 seja percebido como número 5 por daltônicos. Retirado da aplicação móvel executada em um *iPad Air* com tela de 9,7 polegadas

5.4 Filtro para Pranchas de Números 6 a 9

Das pranchas de número 6 até a de número 9, semelhantemente, foi utilizado um mesmo filtro. A Figura 5.3 apresenta a prancha de número 8, a qual possui o número 15 que é identificados por indivíduos normais, porém, os indivíduos daltônicos identificam o número 17. A Figura 5.3 deixa claro então que o problema está no número 7, e é exatamente o local que o aplicativo detecta e mostra para o usuário onde está o problema.

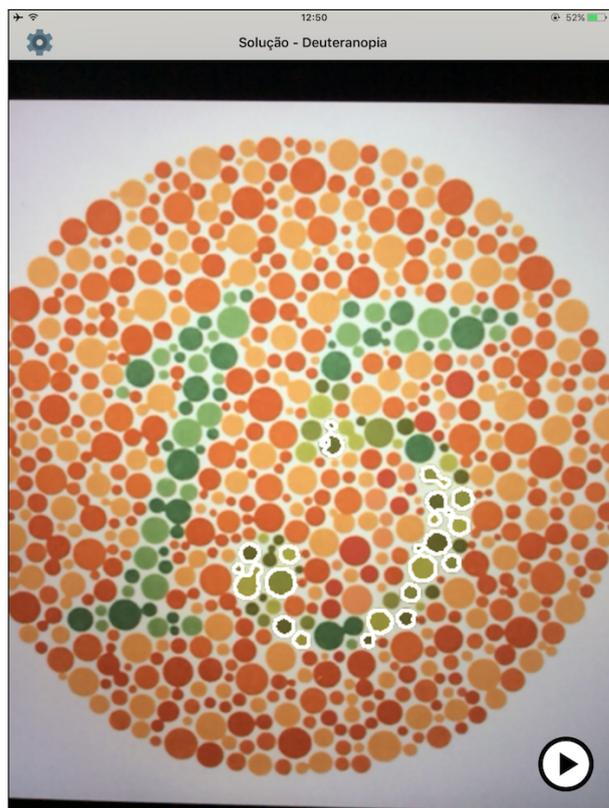


Figura 5.3: Aplicação detecta a parte problemática da imagem da prancha de número 8, responsável por fazer com que o número 15 seja identificado como o número 17 por indivíduos daltônicos. Retirado da aplicação móvel executada em um *iPad Air* com tela de 9,7 polegadas

5.5 Filtro para Pranchas de Números 10 a 13

Para o conjunto de pranchas de número 10 até a de número 13, um novo filtro foi utilizado para a identificação da parte problemática destas imagens. A diferença é que, da prancha de número 10 até a de número 16, indivíduos com discromatopsia não conseguem identificar número algum. Sendo assim, a Figura 5.4 mostra uma imagem da aplicação detectando, na prancha de número 10, o número 2. Como o daltônico não consegue identificar o número nem parte dele, todo o número é identificado e mostrado para o usuário daltônico.

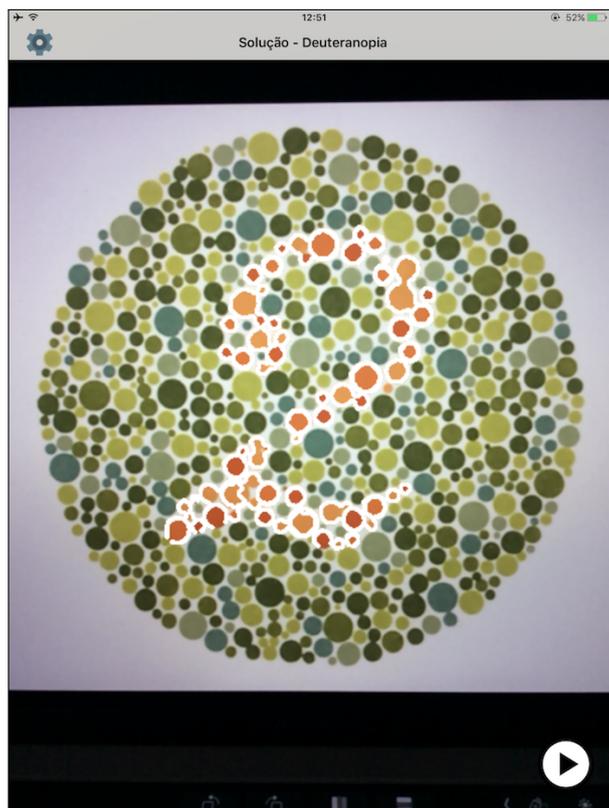


Figura 5.4: Aplicação detecta o número 2 completo, visto que o indivíduo daltônico não consegue identificar número algum. Retirado da aplicação móvel executada em um *iPad Air* com tela de 9,7 polegadas

5.6 Filtro para Pranchas de Números 14 e 17

Um novo filtro foi feito para as pranchas de número 14 e 17 somente, visto que estas duas apresentam combinações de cores distintas das demais pranchas de mesma classificação, necessitando um filtro específico. Sendo assim, a Figura 5.5 mostra a prancha de número 17 e do mesmo modo da Figura 5.4, é identificado o número 73 completo.

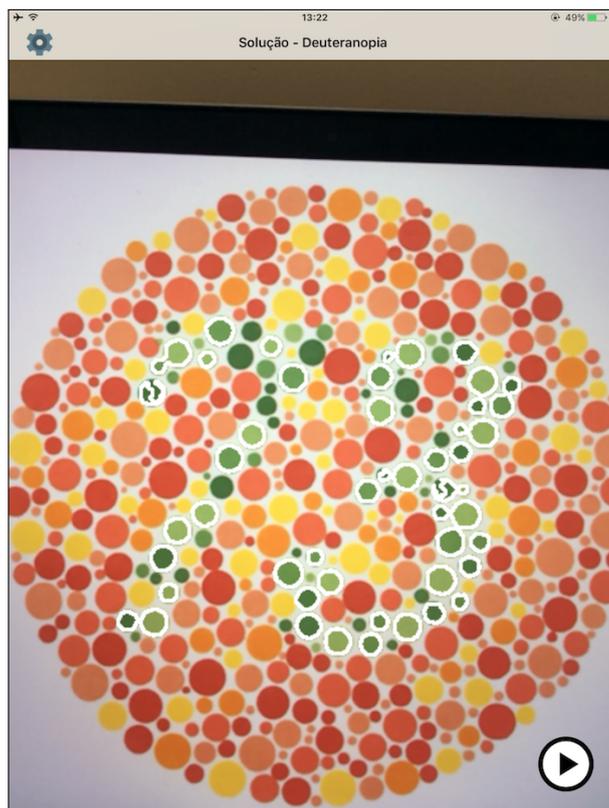


Figura 5.5: Aplicação detecta o número 73 completo, visto que o indivíduo daltônico não consegue identificar número algum. Retirado da aplicação móvel executada em um *iPad Air* com tela de 9,7 polegadas

5.7 Filtro para Pranchas de Números 15 e 16

Outro filtro foi criado para as pranchas de número 15 e 16, sendo que a Figura 5.6 apresenta a prancha de número 15, a qual possui o número 7 que só é identificado por indivíduos normais.

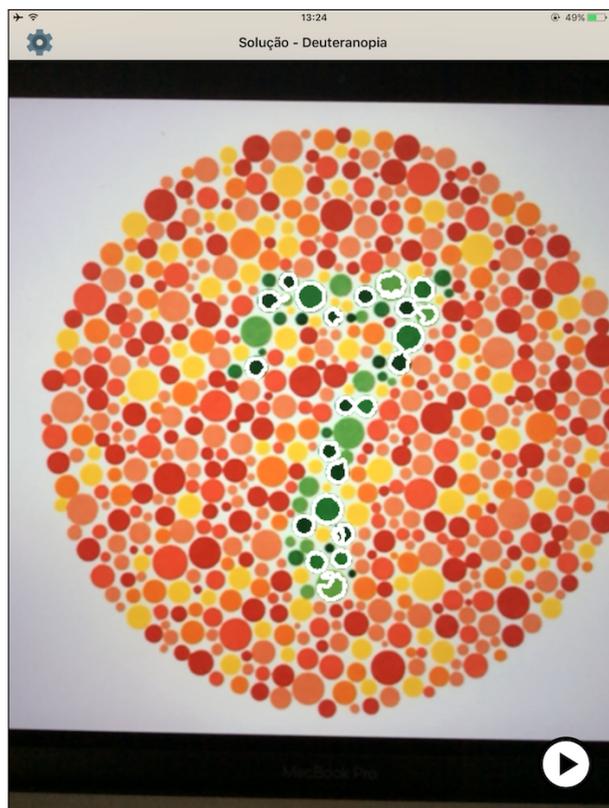


Figura 5.6: Aplicação detecta o número 7 completo, visto que o indivíduo daltônico não consegue identificar número algum. Retirado da aplicação móvel executada em um *iPad Air* com tela de 9,7 polegadas

5.8 Filtro para Pranchas de Números 22 a 25

Por fim, as pranchas de número 22 a 25 são aquelas que possuem o objetivo de classificar o tipo de daltonismo do indivíduo, e isto faz com que pessoas com daltonismos do tipo Deutan identifiquem somente o número da esquerda enquanto que do tipo Protan identificam somente o número da direita.

A Figura 5.7 mostra a imagem da prancha de número 22, a qual contém o número 26. No caso de daltônicos do tipo Deutan, estes só identificam o número 2 e, portanto, a aplicação auxilia-o identificando o número 6, número em que o indivíduo não consegue identificar.

Da mesma forma, porém para a outra parte da imagem, isto é, para o outro número presente, a Figura 5.8 identifica somente o número 2, número no qual indivíduos do tipo Protan não conseguem identificar.



Figura 5.7: Aplicação detecta o número 6 somente, já que indivíduos do tipo Deutan só enxergam o número 2. Retirado da aplicação móvel executada em um *iPad Air* com tela de 9,7 polegadas

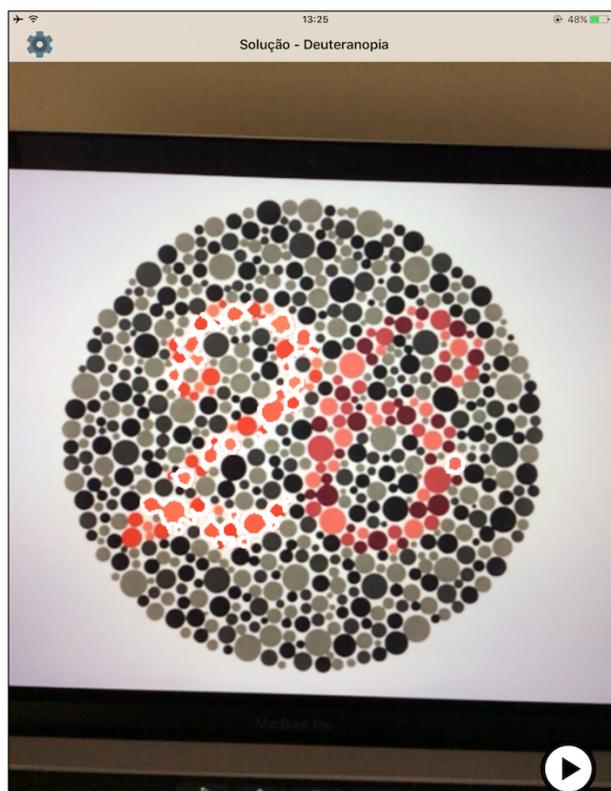


Figura 5.8: Aplicação detecta o número 2 somente, já que indivíduos do tipo Protan só enxergam o número 6. Retirado da aplicação móvel executada em um *iPad Air* com tela de 9,7 polegadas

Capítulo 6

Conclusões e Trabalhos Futuros

O trabalho elaborado propõe uma ferramenta capaz de auxiliar indivíduos que possuem discromatopsia, de forma que o indivíduo seja capaz de saber o que não está sendo distinguido da maneira correta, isto é, todos os elementos que o indivíduo não consegue ou possui certa dificuldade de distinção e percepção de uma ou mais cores. A ferramenta foi elaborada na forma de um aplicativo móvel, desenvolvido na plataforma *iOS* e com o auxílio da biblioteca *OpenCV*, a qual fornece ferramentas por meio de algoritmos de visão computacional capazes de fazer todo o processamento da imagem necessário para a identificação dos elementos do mundo externo que o usuário com daltonismo não consegue distinguir corretamente a cor.

Analisando os resultados apresentados, é possível concluir que o trabalho atingiu o objetivo geral proposto, já que foi desenvolvido o aplicativo móvel capaz de detectar e informar ao usuário daltônico todos os elementos do mundo externo que possuem cores nas quais o usuário possui dificuldade de distinção, isto é, é considerada uma cor problemática para o tipo de daltonismo do usuário, e capaz de realizar o teste de daltonismo mais eficiente e confiável, teste de Ishihara, para fornecer ao usuário do aplicativo um diagnóstico eficiente e prático de daltonismo.

Além da maior contribuição do trabalho que é o aplicativo em si, com suas principais funcionalidades listadas acima, e com o seu objetivo de servir como ferramenta para daltônicos, este trabalho também contribuiu para uma forma de determinação de cores problemáticas para daltônicos baseada na metodologia proposta na Subseção 3.2.6, sendo que esta metodologia foi desenvolvida com base nas imagens do teste de Ishihara, explorando em cada imagem a parte que o daltônico não consegue distinguir, para identificar as cores em que indivíduos com daltonismo possuem dificuldade de percepção e distinção de acordo com o seu tipo de daltonismo, já que cada tipo afeta determinado tipo de cone no olho e resulta em cores diferentes nas quais possui dificuldade de percepção e distinção.

O resultado do trabalho é considerado confiável, visto que foi todo construído com

base no teste mais eficaz de daltonismo existente, e porque foi validado com o próprio teste de Ishihara, sendo possível verificar a veracidade das informações apresentadas nos resultados, quando comparados com os resultados esperados de acordo com a Tabela 2.2, mesmo com diferentes condições de iluminação.

6.1 Trabalhos Futuros

A publicação do aplicativo na loja de aplicativos para sistema *iOS* é um dos trabalhos futuros, já que o mesmo ainda não se encontra publicado devido a desenvolvimentos futuros pendentes, tais como melhorias na usabilidade, implementação de uma tela contendo informações acerca do daltonismo e tradução dos nomes das 8 cores segmentadas para o português, visando torná-lo uma ferramenta completa para daltônicos.

O desenvolvimento do aplicativo para dispositivo móvel criado pode ser também adaptado para óculos inteligentes, utilizando a metodologia proposta neste trabalho e adaptando somente a linguagem de programação usada e a questões de interação com o usuário e navegação no aplicativo. Óculos inteligentes são óculos equipados com um mini projetor e com uma câmera, que permite para o seu usuário uma experiência de realidade aumentada [50]. Desta forma, seria mais prático para o usuário receber informações sobre objetos com cores consideradas problemáticas, de acordo com o seu tipo de daltonismo, sem precisar utilizar um *tablet* ou *smartphone*, já que as informações seriam apresentadas em tempo real e em forma de realidade aumentada, trazendo mais conforto e comodidade para o indivíduo portador de discromatopsia e que utiliza esta ferramenta de auxílio.

Outra proposta para um trabalho futuro seria a implementação da ferramenta de auxílio para os tipos mais raros de daltonismos, como a Tritanopia e monocromatismo. Esta implementação exige pesquisas mais avançadas, pois deve ser levado em consideração a carência de pesquisas para esses tipos mais raros de discromatopsias, já que o teste de Ishihara é eficaz somente para os tipos Protan e Deutan de daltonismo. Para esta implementação, seria necessário identificar a faixa de cores consideradas problemáticas para esses tipos e encontrar um mecanismo de validação destas cores identificadas.

Portanto, um trabalho futuro pode ser definido pela implementação para todos os tipos, incluindo os mais raros, de daltonismos em um dispositivo mais adequado para sua utilidade, como um óculos inteligente, adaptando o código para ser utilizado no óculos, capaz de mostrar informações geradas em tempo real utilizando realidade aumentada.

Outro importante trabalho a ser realizado é a validação da ferramenta com indivíduos daltônicos por meio de pesquisa com pessoas com os mais diversos tipos de daltonismo, validando o aplicativo e recebendo feedback dos mesmos para melhorias na ferramenta.

Referências

- [1] Ishihara, S.: *The series of plates designed as tests for colour-blindness: 38 plates ed.* Kanehara Trading Inc., 1951. ix, 6, 7, 8, 9, 11, 16, 26, 29
- [2] *Espaço de cores rgb.* <http://www.ufrgs.br/engcart/PDASR/formcor.html#2>, acesso em 30-11-2017. ix, 12
- [3] Sulsoft: *Envi - guia do envi em português*, 2000. <http://www.sulsoft.com.br>. ix, 13
- [4] Simões, Alexandre da Silva: *Segmentação de imagens por classificação de cores: uma abordagem neural.* Tese de Doutorado, Universidade de São Paulo, 2000. ix, 13, 14
- [5] Feitosa-Santana, Claudia, Nestor Norio Oiwa, Marcelo Fernandes da Costa, Klaus Bruno Tiedemann, Luiz Carlos de Lima Silveira e Dora Fix Ventura: *Espaço de cores.* Psicologia USP, 17:35 – 62, 2006, ISSN 0103-6564. http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-65642006000400003&lng=en&nrm=iso, acesso em 20-10-2017. 1
- [6] Kaiser, Peter K. e Robert M. Boynton: *Human color vision. 2nd ed.* Washington: Optical Society of America, 1986. 1
- [7] Pedrosa, Israel: *Leonardo da vinci e a teoria das cores. in: pedrosa i, editor. da cor à cor inexistente.* páginas 37–48, Rio de Janeiro, 1982. 1
- [8] Eisberg, Robert M. e Lawrence S. Lerner: *Física. fundamentos e aplicações, v.4.* São Paulo, 1983. McGraw-Hill. 1
- [9] Halliday, D e R Resnick: *Fundamentos de física, v. 4.* Rio de Janeiro: Livros Técnicos e Científicos, 1991. 1
- [10] Casarin, F. C. F.: *O daltonismo: um exemplo de herança ligada ao cromossomo x.* Tese de Mestrado, Universidade Federal do Paraná, Cruzeiro do Oeste, 2015. 1
- [11] Unonius, Nichard, Michel Eid Farah e Juliana M. Ferraz Sallum: *Classificação diagnóstica dos portadores de doenças degenerativas de retina, integrantes dos grupos Retina São Paulo e Retina Vale do Paraíba.* Arquivos Brasileiros de Oftalmologia, 66:443 – 448, agosto 2003, ISSN 0004-2749. http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0004-27492003000400009&nrm=iso, acesso em 22-10-2017. 1

- [12] Jefferson, Luke e Richard Harvey: *Accommodating color blind computer users*. Em *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility*, Assets '06, páginas 40–47, New York, NY, USA, 2006. ACM, ISBN 1-59593-290-9. <http://doi.acm.org/10.1145/1168987.1168996>. 1
- [13] *New studies rely on the internet for help treating cancer patients*, Jun 2016. <http://www.foxnews.com/health/2016/06/07/new-studies-rely-on-internet-for-help-treating-cancer-patients.html>, acesso em 30-11-2017. 2
- [14] iClinic: *O poder da tecnologia aplicada à saúde*, Oct 2017. <http://blog.iclinic.com.br/o-poder-da-tecnologia-aplicada-a-saude/>, acesso em 01-12-2017. 2
- [15] Souza, Jacqueline Fontes de, Francisco Brandão Gonçalves, Valéria Argôlo Rosa de Queiroz e Rodrigo Santos de Queiroz: *Avaliação de um aplicativo para auxílio à tomada de decisão de mobilizar pacientes críticos*. *Saúde*, 2016(11):1. 2
- [16] Reinicke, Matheus, Maciel Hogenn, Douglas Felipe Hoss, Gustavo Carletto e Eliana Vogel Jaeger: *A utilização de tecnologias no auxílio ao tratamento de diabetes melitus: Um estudo realizado com acadêmicos do centro de educação do alto vale do itajaí*. REAVI-Revista Eletrônica do Alto Vale do Itajaí, 2(2):165–168, 2013. 2
- [17] Vespucci, K M: *Daltônicos ao volante. disponível em: http://www.sinaldetransito.com.br/artigos/daltonicos_ao_volante.pdf.* acesso em novembro de. 2017. 2, 4
- [18] Soares, Renato: *Avaliação dos condutores portadores de discromatopsia congênita na percepção cromática da sinalização viária. dissertação de mestrado em transportes, publicação t.dm – 008a/2009. departamento de engenharia civil e ambiental, faculdade de tecnologia, universidade de.* 2009. 4, 7
- [19] Crepaldi, T. O. M.: *Desempenho de acromatas no preenchimento perceptual*. Tese de Mestrado, Faculdade de Ciências da Saúde, Universidade de Brasília, 2003. 4
- [20] Krill, Alex E. e Desmond B. Archer: *Hereditary retinal and choroidal diseases: Evaluation*. 1972. 5
- [21] Merin, Saul: *Inherited eye diseases: Diagnosis and clinical management*. CRC Press, 2013. 5
- [22] Costa, G. L. de S. C.: *Daltonismo e suas consequências*. 2011. Disponível em: contatogo.blogspot.com/2011/09/daltonismo-e-suas-consequencias.html. 5
- [23] Sato, Mário Teruo, Alfredo Vidal Moreira, Daniel Roncglio Guerra, Ana Cristina Alvarez de Carvalho e Carlos Augusto Moreira Júnior: *Discromatopsias congênitas e condução de veículos. vol. 65. arquivo brasileiro de oftalmologia. são paulo*. 2002. 5
- [24] Bruni, Lígia Fernanda e Antonio Augusto Velasco Cruz: *Sentido cromático: Tipos de defeitos e testes de avaliação clínica*. *Arquivos Brasileiros de Oftalmologia*, 69(5):766–775, 2006. 5

- [25] *Facts about color blindness*. U.S. Department of Health and Human Services, Feb 2015. https://nei.nih.gov/health/color_blindness/facts_about. 5
- [26] *Color vision deficiency - genetics home reference*. <https://ghr.nlm.nih.gov/condition/color-vision-deficiency#inheritance>. 6
- [27] Fletcher, Robert e Janet Voke: *Defective colour vision, fundamentals, diagnosis and management*. CRC Press, 1985. 6
- [28] Fernandes, Luciene Chaves e Lúcia Carvalho de Ventura Urbano: *Eficiência dos testes cromáticos de comparação na discromatopsia hereditária: relato de casos*. Arquivos Brasileiros de Oftalmologia, 71(4):585–588, 2008. 6
- [29] Birch, Jennifer: *Efficiency of the ishihara test for identifying red-green colour deficiency*. Ophthalmic and Physiological Optics, 17(5):403–408, 1997. 6
- [30] Hardy, Le Grand H, Gertrude Rand e M Catherine Rittler: *Tests for detection and analysis of color blindness: an evaluation of the ishihara test*. Archives of Ophthalmology, 34(4):295–302, 1945. 6
- [31] Trivedi, M. M. e A. Rosenfeld: *On making computers see*. IEEE Trans Syst Man Cybern, 19(6):1333–1335, nov. 1989. 11
- [32] Marr, David: *Vision. new york: W h. H Freeman and Company*, 1982. 11
- [33] Forsyth, David e Jean Ponce: *Computer vision: a modern approach*. Upper Saddle River, NJ; London: Prentice Hall, 2011. 12
- [34] Feitosa, Rafael Divino Ferreira: *Modelos matemáticos para redução do espectro provável e detecção de tons de pele humana em imagens coloridas representadas nos espaços de cores rgb e hsv*. Tese de Mestrado, 2015. 12, 13
- [35] Kakumanu, Praveen, Sokratis Makrogiannis e Nikolaos Bourbakis: *A survey of skin-color modeling and detection methods*. Pattern recognition, 40(3):1106–1122, 2007. 12
- [36] *Modelo de cores hsv*. <http://www.ufrgs.br/engcart/PDASR/formcor.html#4>. 13
- [37] Manjare, Shenal e S. R. Chougule: *Skin detection for face recognition based on hsv color space*. International Journal of Engineering Sciences & Research Technology, 2(7):1883, 2013. 13
- [38] Ballard, D e C Brown: *Computer vision*. Prentice Hall, 1982. 13
- [39] *OpenCV library*. <https://opencv.org/>. 14
- [40] *About - opencv*. <https://opencv.org/about.html>. 14
- [41] *iOS*. <https://www.apple.com/ios/>. 15, 16
- [42] Williams, Rhiannon: *Apple iOS: a brief history*, Sep 2015. <http://www.telegraph.co.uk/technology/apple/11068420/Apple-iOS-a-brief-history.html>. 15

- [43] Apple: *What's new in the app store*. <https://developer.apple.com/app-store/>. 15
- [44] Apple: *From code to customer*. <https://developer.apple.com/programs/>. 15
- [45] *Objective-c*. <https://developer.apple.com/documentation/objectivec>. 15
- [46] *Swift standard library*. <https://developer.apple.com/documentation/swift>. 15
- [47] *Number of mobile phone users worldwide 2013-2019*. <https://www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/>. 15
- [48] *Tablet users worldwide 2013-2020 | statistic*. <https://www.statista.com/statistics/377977/tablet-users-worldwide-forecast/>. 15
- [49] *Mobile and tablet internet usage exceeds desktop for first time worldwide*. <http://gs.statcounter.com/press/mobile-and-tablet-internet-usage-exceeds-desktop-for-first-time-worldwide>. 15
- [50] *Glass*. <https://www.x.company/glass/>, acesso em 25-11-2017. 16, 56
- [51] Ieee: *Ieee standard glossary of software engineering terminology*. 1990. 17
- [52] Sommerville, Ian: *Engenharia de software-8ª edição (2007)*. Ed Person Education. 17
- [53] Apple: *What's new in xcode 9*. <https://developer.apple.com/xcode/>. 22, 24
- [54] *Miscellaneous image transformations*. https://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous_transformations.html. 29
- [55] *Eroding and dilating*. https://docs.opencv.org/2.4/doc/tutorials/imgproc/erosion_dilatation/erosion_dilatation.html, acesso em 01-12-2017. 37
- [56] Oliveira, Luiz Eduardo S.: *Processamento de imagens. morfologia matemática binária*. <http://www.inf.ufpr.br/lesoliveira/download/morfologia.pdf>. 37, 38
- [57] *Operations on arrays*. https://docs.opencv.org/3.0-beta/modules/core/doc/operations_on_arrays.html, acesso em 01-12-2017. 39
- [58] *Structural analysis and shape descriptors*. https://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html?highlight=findcontours#findcontours, acesso em 01-12-2017. 41
- [59] *Understanding k-means clustering*. OpenCV 3.0.0- dev documentation. https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_ml/py_kmeans/py_kmeans_understanding/py_kmeans_understanding.html#kmeans-clustering-understanding. 44