

UNIVERSIDADE DE SÃO PAULO

Instituto de Matemática e Estatística

**Ferramenta de acessibilidade adaptável aos daltônicos e às
redes móveis**

Alex Takata

Orientador: Prof. Alfredo Goldman vel Lejbman

São Paulo

janeiro de 2015

Lista de Figuras

Figura 1 - Esquema da estrutura do olho humano	6
Figura 2 - A estrutura celular da retina.	7
Figura 3 - Gráfico da absorção de luz por comprimento de onda.....	8
Figura 4 - Imagem de demonstração do teste de Ishihara - Numeral	11
Figura 5 – Imagem de demonstração do teste de Ishihara - Trajetória.....	11
Figura 6 – Tipos de imagens do teste de Ishihara.....	12
Figura 7 - Arquitetura Android.....	13
Figura 8 - Paleta de cores após aplicação de filtro de acessibilidade	16
Figura 9 - Gráfico do tempo de execução x quantidade de pixels da imagem no servidor .	19
Figura 10 - Gráfico do tempo de execução x quantidade de pixels da imagem no dispositivo móvel.....	21
Figura 11 - Tela Inicial	25
Figura 12 - Menu Principal	26
Figura 13 - Teste de Ishihara	26
Figura 14 - Menu de configurações	27
Figura 15 – Função acessibilidade.....	28

Lista de Tabelas

Tabela 1 - Proporção da ocorrência da dicromacia e tricromacia anômala	10
Tabela 2 - Tempo de processamento da imagem no servidor.....	18
Tabela 3 - Tempo de processamento da imagem no dispositivo móvel	21
Tabela 4 - Informações das imagens utilizadas nos testes de <i>offloading</i>	29
Tabela 5 - Informações obtidas das redes móveis disponíveis	29
Tabela 6 – Dados para a análise de quando realizar <i>offloading</i>	30
Tabela 7 - Informações do processamento das imagens	31

Sumário

Parte Objetiva	1
1 Introdução	1
2 Objetivos	2
3 Trabalhos relacionados	3
3.1 Visocor – Sistema de Acessibilidade Visual	3
3.2 Uma Análise do Impacto da Qualidade da Internet Móvel na Utilização de <i>Cloudlets</i>	3
3.3 Ferramentas existentes	4
4 Conceitos e tecnologias estudadas	6
4.1 Percepção de cores	6
4.2 Daltonismo	8
4.2.1 Tipos de daltonismo	8
4.3 Teste de Ishihara	10
4.4 Android	12
4.5 <i>Offloading</i>	14
5 A ferramenta	15
5.1 Filtro de acessibilidade	15
5.2 Análise para <i>Offloading</i>	16
5.2.1 Estimativa do tempo de processamento da imagem	16
5.2.2 Estimativa do tempo de transmissão da imagem	22
5.2.3 Relação memória disponível x memória a ser alocada	23
6 Conclusão	25

6.1	Resultados.....	25
6.2	Trabalhos Futuros	32
7	Referência bibliográfica.....	33
	Parte Subjetiva.....	35
1	Desafios e frustrações	35
2	Disciplinas mais relevantes para o trabalho.....	35

Parte Objetiva

1 Introdução

O daltonismo é um distúrbio na visão caracterizada por reduzir parcialmente ou completamente a capacidade de distinguir algumas cores. Tal perturbação atinge 8% da população masculina e 0,5% da população feminina [1]. Esta parcela da população enfrenta diversas dificuldades em realizar tarefas simples do dia a dia, como ver mapas, gráficos e tabelas onde são utilizadas as cores em suas legendas.

Com o mundo da tecnologia evoluindo constantemente percebemos que algumas dificuldades podem ser amenizadas e soluções de acessibilidade encontradas para aqueles que possuem algum tipo de distúrbio. Analisando o universo da mobilidade, os dispositivos móveis capazes de processar informações estão sendo inseridos cada vez mais no dia a dia das pessoas. Utilizando esses dispositivos como ferramentas de acessibilidade é possível melhorar a qualidade visual de imagens de pessoas que tenham algum nível de daltonismo.

No entanto, apesar dos avanços tecnológicos, os dispositivos móveis apresentam certas limitações como capacidade de processamento, memória e armazenamento de dados, quando comparados com os computadores. Isso torna algumas operações inviáveis de serem realizadas nos dispositivos.

Uma estratégia para tornar estas operações viáveis nos dispositivos móveis é utilizar a técnica de *offloading*, que permite transferir o processamento de dados e tarefas complexas para outra infraestrutura com maior capacidade de processamento e armazenamento, como *Cloudlet* ou nuvem pública.

2 Objetivos

Este trabalho tem como objetivos o estudo das anomalias na percepção de cores e o desenvolvimento de uma ferramenta de acessibilidade adaptável aos diferentes tipos e graus de daltonismo e às redes móveis.

Procurou-se desenvolver uma ferramenta para a plataforma Android que utilizasse filtros de acessibilidade para transformar contrastes difíceis de serem identificados pelos daltônicos em contrastes visíveis, tendo como base o trabalho de conclusão de curso realizado pelos ex-alunos do BCC A. Asato e R. Gonçalves em 2009.

A ferramenta utilizaria informações das redes móveis e do dispositivo para avaliar o melhor local de processamento da imagem, e se necessário, transferir a tarefa de transformação da imagem para um servidor com melhor capacidade computacional.

Além disso, outra funcionalidade disponível na ferramenta seria o teste de Ishihara para diagnosticar a presença de distúrbios na percepção de cores do usuário, e com o resultado configurar automaticamente o filtro de acessibilidade.

3 Trabalhos relacionados

3.1 Visocor – Sistema de Acessibilidade Visual

Visocor[2] é uma ferramenta de acessibilidade para as pessoas com deficiência visual de cores desenvolvida no trabalho de conclusão de curso do BCC dos alunos André Shoji Asato e Rafael de O. Lopes em 2009.

O software foi desenvolvido como um plugin open-source para o Compiz utilizando a linguagem de programação C, e as APIs fornecidas pelo Compiz e pelo OpenGL, permitindo o seu funcionamento em diversas plataformas.

A ferramenta possui uma funcionalidade de acessibilidade que realiza o processamento gráfico em tempo real utilizando um filtro parametrizável que tem como finalidade permitir a percepção do contraste vermelho/verde. Além desta função, o Visocor permite que o usuário consiga destacar uma cor da tela.

3.2 Uma Análise do Impacto da Qualidade da Internet Móvel na Utilização de *Cloudlets*

O trabalho "Uma análise do impacto da qualidade da internet móvel na utilização de *cloudlets*" de P. Costa, P. Rego, E. Coutinho, F. Trinta, J. Souza [3], apresenta uma avaliação da qualidade da internet móvel em três capitais do Nordeste do Brasil e um estudo comparativo entre o uso de *cloudlets* e a nuvem pública para realizar *offloading*, conectadas via diferentes redes sem fio.

Para este estudo foi desenvolvido duas aplicações para a plataforma Android. A primeira, Netester, exibe informações da rede móvel. É apresentado na tela do dispositivo o tipo de rede conectada, a quantidade de dados enviados, a quantidade de pacotes perdidos, as taxas de transferência dos dados, os intervalos de chegada dos pacotes, e os resultados do ping do protocolo TCP e UDP. A segunda aplicação, BenchImage, aplica efeitos de Cartoonizer¹ em

¹ Cartoonizer é um efeito que aplica quatro filtros na imagem, dentre eles o desfoque gaussiano e o ColorDodgeBlend.

imagens. O processamento da imagem pode ser realizado no dispositivo móvel, no *Cloudlet* ou Nuvem física. Após o processamento da imagem, a mesma é enviada ao dispositivo.

Além de avaliar a qualidade da rede móvel, os autores discutem os fatores que impactam na tomada de decisão de quando e onde fazer *offloading*, e mostram que os fatores de maiores pesos para a decisão são: a quantidade de dados a serem enviados, a complexidade das tarefas a serem executadas, e a vazão de rede.

3.3 Ferramentas existentes

Após pesquisa realizada na Google Play, loja de aplicativos para o sistema Android, foi possível constatar que dentre as funcionalidades existentes as mais evidentes nos aplicativos para os dispositivos móveis são o teste de Ishihara e suas variações, utilizados para identificar o tipo de daltonismo que a pessoa possui. Outra funcionalidade que vemos constantemente nas ferramentas de acessibilidade aos daltônicos é a de destaque de contrastes não perceptíveis aos diferentes tipos de daltonismo através da aplicação de filtros em imagens.

Existem também aplicativos onde é possível identificar a cor de determinado ponto de uma imagem ou utilizar a câmera do dispositivo para identificar a cor predominante de uma região em tempo real, e aplicativos que simulam a visão de um daltônico transformando as cores de imagens do dispositivo móvel.

São apresentadas abaixo algumas ferramentas com as funcionalidades descritas acima:

Vischeck Daltonize Tool

Vischeck [\[4\]](#) é uma ferramenta de acessibilidade que oferece duas funcionalidades. A primeira permite que o usuário selecione um tipo de daltonismo dentre as opções: Protanopia, Deuteranopia e Tritanopia, e simula a visão de uma pessoa com a opção escolhida.

A segunda funcionalidade transforma contrastes não visíveis aos daltônicos em contrastes visíveis, através do ajuste do contraste vermelho/verde e azul/amarelo, e o brilho de imagens que pode ser realizado pelo usuário.

Além de oferecer os serviços online, está disponível também a ferramenta em forma de plugin para Adobe Photoshop e ImageJ.

HueVue

Aplicativo desenvolvido para iOS que auxilia os daltônicos na identificação de cores que aparecem na tela do dispositivo móvel. É possível tirar foto com a câmera para identificar as cores ou selecionar uma imagem que esteja armazenada no aparelho. [\[5\]](#)

Chromatic Vision Simulator

Ferramenta disponível para os sistemas operacionais iOS e Android, e para Web, que simula a visão dos daltônicos a partir de uma imagem selecionada. [\[6\]](#)

Colorblind Vision

Ferramenta para o sistema Android que além de apresentar informações sobre o que é daltonismo, os tipos existentes e estatísticas, possui as funcionalidades: simulação da visão de um daltônico, teste para verificar se o usuário possui daltonismo, e extrator de cores, que destaca apenas a cor selecionada de uma imagem em seu menu. [\[7\]](#)

4 Conceitos e tecnologias estudadas

4.1 Percepção de cores

O olho humano apesar de possuir um tamanho pequeno é um órgão muito complexo. Para o homem, os olhos são instrumentos de desenvolvimento do aprendizado, pensamento e de comunicação com outros seres humanos. É com a visão que se pode captar e interpretar os diversos elementos de um ambiente.

O sistema visual é composto pelos olhos, nervos ópticos e pelo cérebro. Os olhos, órgãos sensoriais, transformam a luz em um sinal neural; os nervos ópticos, por sua vez, transportam este sinal até o cérebro, que processa e extrai as informações necessárias. No cérebro, a percepção visual ocorre no lóbulo occipital, que realiza a interação desta percepção com outras informações. Os lóbulos parietais e temporais são os responsáveis pela compreensão do que se vê [14].

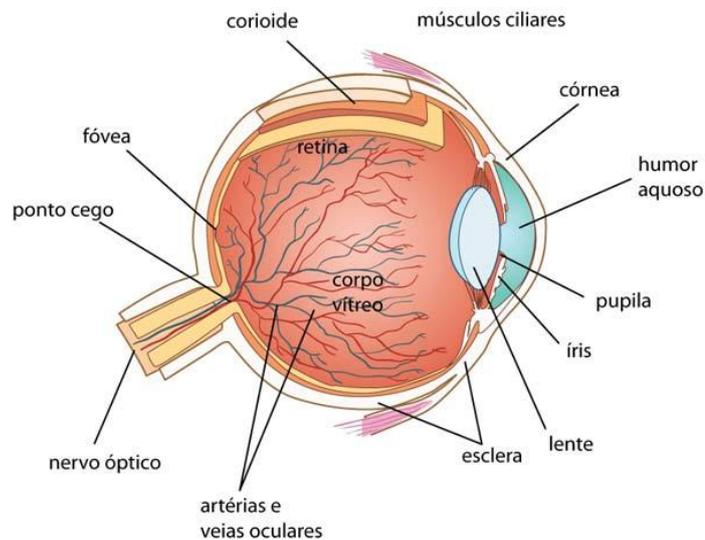
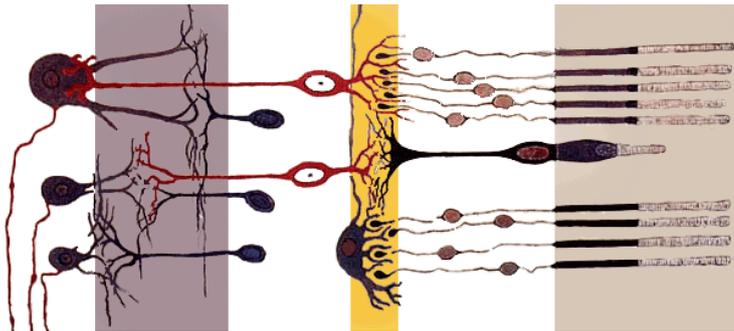


Figura 1 - Esquema da estrutura do olho humano (Fonte: [13])

O olho humano, que é aproximadamente esférico com diâmetro em torno de 2,5 cm e peso de 7 g, fica acondicionado dentro de uma cavidade óssea. Possui uma estrutura complexa formada por diversos elementos como córnea, íris e retina. Neste último, estão localizados os fotorreceptores, células responsáveis pela captação de estímulos luminosos, que se subdividem em porção periférica e central.

Na porção periférica predominam os fotorreceptores conhecidos como bastonetes, enquanto na porção central os fotorreceptores conhecidos como cones são mais abundantes.

A quantidade de bastonetes existentes no olho humano é de aproximadamente 120 milhões, já a quantidade de cones existentes é de aproximadamente 6 milhões.



*Figura 2 - A estrutura celular da retina. À direita, um cone entre dois grupos de bastonete
(Fonte: [11])*

Os bastonetes tem alta sensibilidade acromática à luz, característica escotópica, ou seja, não possuem a capacidade de distinguir as cores, porém são importantes para a visão de objetos em ambientes com baixa luminosidade.

Os cones são menos sensíveis à luz, mas possuem a capacidade de discriminarem os diferentes comprimentos de onda, característica chamada de fotópica, possibilitando a visão em cores [14].

Existem três tipos diferentes de cones, cada um contendo um tipo de pigmento fotossensível, classificados conforme o comprimento de onda pelo qual são estimulados. Os cones estimulados por comprimentos de onda longos (tipo L), em torno dos 670 nm, são sensíveis à cor vermelha. Os sensíveis aos comprimentos de onda médios (tipo M), na faixa dos 540 nm, são sensíveis ao verde. E os cones sensíveis aos comprimentos de onda curtos (tipo S), por volta dos 440 nm, são sensíveis ao azul.

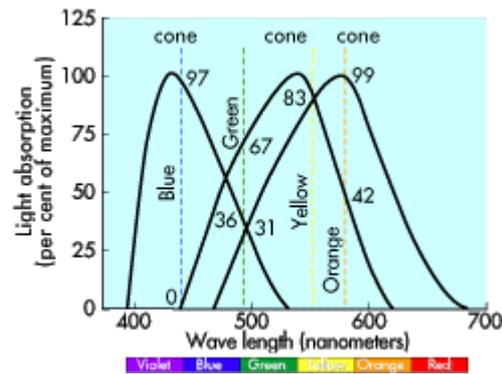


Figura 3 - Gráfico da absorção de luz por comprimento de onda (Fonte: [12])

4.2 Daltonismo

Estudos mostram que a deficiência na visão em relação à percepção de cores atinge 8% da população masculina e 0,5% da população feminina. Esta deficiência, também conhecida como daltonismo, se manifesta como uma dificuldade em distinguir cores que uma pessoa com visão normal conseguiria.

O primeiro intelectual a estudar com seriedade este distúrbio foi o inglês John Dalton, físico, químico e meteorologista, quem realizou uma autoanálise de sua deficiência para cores em 1798, e posteriormente foi homenageado com o termo daltonismo.

Tal perturbação pode ser adquirida devido a uma lesão nos órgãos responsáveis pela visão, porém é provocada principalmente por genes recessivos localizados no cromossomo X. Como os homens possuem apenas um cromossomo X, eles são mais afetados do que as mulheres, que possuem dois cromossomos X [15].

4.2.1 Tipos de daltonismo

Monocromacia

Deficiência visual causada pela ausência de dois ou três tipos de cones, que faz com que as pessoas não consigam distinguir qualquer cor, enxergando tudo em tons de cinza. Classificada em:

Monocromacia Rod: Também conhecida como acromatopsia, ocorre quando os bastonetes da retina estão presentes e funcionais, porém os três tipos de cones não estão funcionais ou não estão presentes. Homens e mulheres são igualmente propensos a terem este tipo de daltonismo, pois este não é ligado ao sexo.

Monocromacia Cone: Ocorre quando apenas um tipo de cone está presente na retina. Sendo assim, ela pode ser classificada como:

S-Monocromacia: Presente apenas o cone do tipo S.

M-Monocromacia: Presente apenas o cone do tipo M.

L-Monocromacia: Presente apenas o cone do tipo L.

Dicromacia

A dicromacia ocorre quando um tipo de cone não está presente na retina. É possível classificá-la em:

Protanopia: Ausência dos cones do tipo L, fazendo com que as pessoas sejam menos sensíveis à luz vermelha, com isto existe uma dificuldade em distinguir os pares de cores azul/verde e vermelho/verde.

Deuteranopia: Ausência dos cones do tipo M, dificultando a distinção das cores vermelho/verde, roxo/azul e alguns tons de cinza.

Tritanopia: Ausência dos cones do tipo S. Pessoas com esta anomalia possuem dificuldade em distinguir o azul do verde, e o amarelo do violeta.

Tricromacia Anômala

Na tricromacia anômala os três tipos de cones estão presentes na retina, porém um deles possui alguma alteração, fazendo com que o daltonismo apareça em diferentes intensidades. É possível classificá-la conforme abaixo:

Protanomalia: Ocorre quando há uma anomalia nos cones do tipo L tornando a pessoa menos sensível à luz vermelha. A percepção de cores é semelhante à protanopia, porém mais suave.

Deuteranomalia: Ocorre quando há uma anomalia nos cones do tipo M, no qual passam a ter o pico de onda bem próximo ao do cone do tipo L. Embora os indivíduos com deuteranomalia provavelmente não possam visualizar vermelhos e verdes da mesma forma que as pessoas com visão normal, muitas vezes podem distinguir entre os tons de vermelhos e verdes com relativa exatidão.

Tritanomalia: Uma forma atenuada da tritanopia. Ocorre quando há uma anomalia nos cones do tipo S, sensíveis à luz de comprimentos de onda curtos, dificultando a distinção das cores azul/verde e amarelo/violeta [15].

Tipo de daltonismo		Homens	Mulheres
Monocromacia	Monocromacia Rod	Raro	Raro
	Monocromacia Cone	Raro	Raro
Dicromacia	Protanopia	1%	0,01%
	Deutaranopia	1,5%	0,01%
	Tritanopia	0,008%	0,008%
Tricromacia anômala	Protanomalia	1%	0,01%
	Deuteranomalia	5%	0,4%
	Tritanomalia	Raro	Raro

Tabela 1 - Proporção da ocorrência da dicromacia e tricromacia anômala

4.3 Teste de Ishihara

Criado pelo professor Dr. Shinobu Ishihara da Universidade de Tóquio, o teste de Ishihara tem como objetivo diagnosticar de forma rápida e precisa a deficiência congênita para a visão em cores, porém, também pode ser utilizada para verificar problemas adquiridos na percepção de cores.

O teste consiste na exibição de uma série de imagens, onde cada uma possui círculos desordenados e de diversos tamanhos escondendo as linhas de um numeral conforme exemplo da *Figura 4*, ou escondendo uma trajetória de uma letra X até outra letra X conforme exemplo da *Figura 5*, sendo perceptível apenas devido aos contrastes de cores de cada círculo [8].

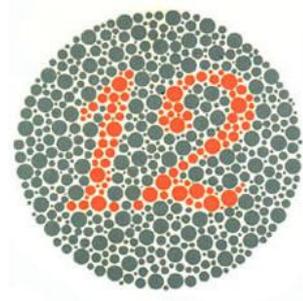


Figura 4 - Imagem de demonstração do teste de Ishihara - Numeral

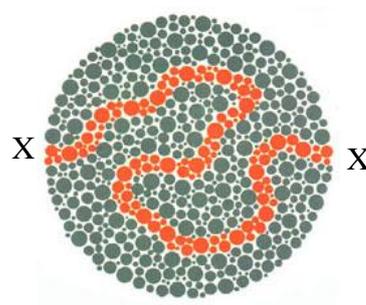


Figura 5 – Imagem de demonstração do teste de Ishihara - Trajetória

As imagens do teste de Ishihara podem ser classificadas como:

- Imagens de demonstração: Visíveis por todos os observadores, devido ao alto contraste em relação ao fundo.
- Imagens de transformação: Pessoas com deficiência na percepção de cores visualizam uma figura diferente das pessoas com visão normal.
- Imagens mascaradas: Apenas as pessoas com visão normal conseguem visualizar a figura, pois apresentam uma cor próxima ou no eixo de confusão do indivíduo portador de daltonismo.
- Imagens escondidas: Apenas as pessoas que possuem algum tipo de deficiência na percepção de cores são capazes de identificar a imagem.
- Imagens de diagnóstico: Possuem duas figuras mascaradas, onde uma é identificada pelos protanômalos e outra pelos deuteranômalos.

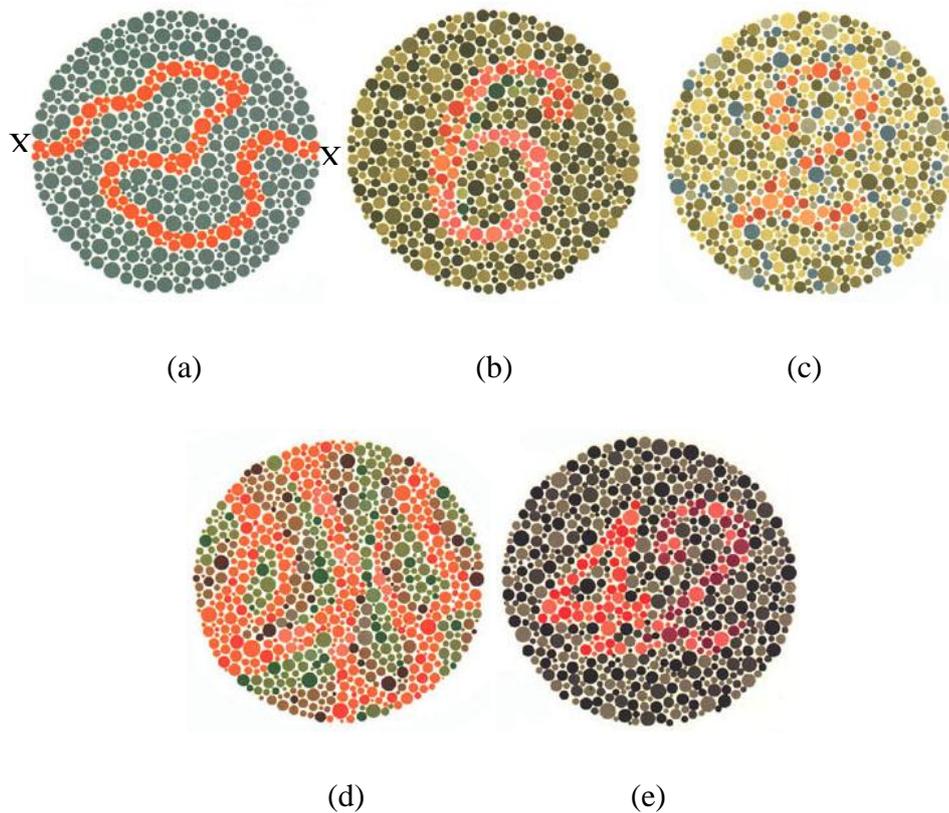


Figura 6 – Tipos de imagens do teste de Ishihara: (a) Imagem de demonstração, (b) Imagem de transformação, (c) Imagem mascarada, (d) Imagem escondida, (e) Imagem de diagnóstico

4.4 Android

O Android é um sistema operacional para dispositivos móveis baseado no núcleo do Linux desenvolvido inicialmente pela empresa *Android Inc.* que foi posteriormente adquirida pela Google em 2005. Com a parceria de empresas fabricantes de hardware, software e operadoras de telefonia móvel, no final de 2008 foi vendido o primeiro celular utilizando o Android como sistema operacional [9][10][11].

Devido a sua abertura para modificações, os fabricantes de dispositivos podem trabalhar livremente nas suas personalizações, pois o código do sistema está sob a Licença Apache. Os aplicativos desenvolvidos para o sistema Android são codificados utilizando a linguagem Java, o que facilita a criação de novas aplicações. Com o amadurecimento do sistema e as oportunidades apresentadas aos fabricantes, desenvolvedores e usuários fizeram com que o

mercado adotasse o Android em grande escala. É possível procurar por boas práticas de programação dos aplicativos para fornecer ao usuário uma melhor experiência com os aplicativos.

Para desenvolver um aplicativo é necessário utilizar o kit disponibilizado em seu site que contém as bibliotecas, o debugger, exemplos, tutoriais e o emulador para testar o aplicativo desenvolvido. De tempos em tempos é disponibilizada uma atualização do Sistema Operacional com melhorias no desempenho, incremento de recursos como segurança e personalizações. Na construção dos aplicativos é possível especificar quais os requisitos mínimos necessários para a instalação do aplicativo, restringindo os aparelhos alvos e fazendo com que uma aplicação não seja prejudicada por causa das especificações do aparelho.

A arquitetura do sistema Android é a seguinte:

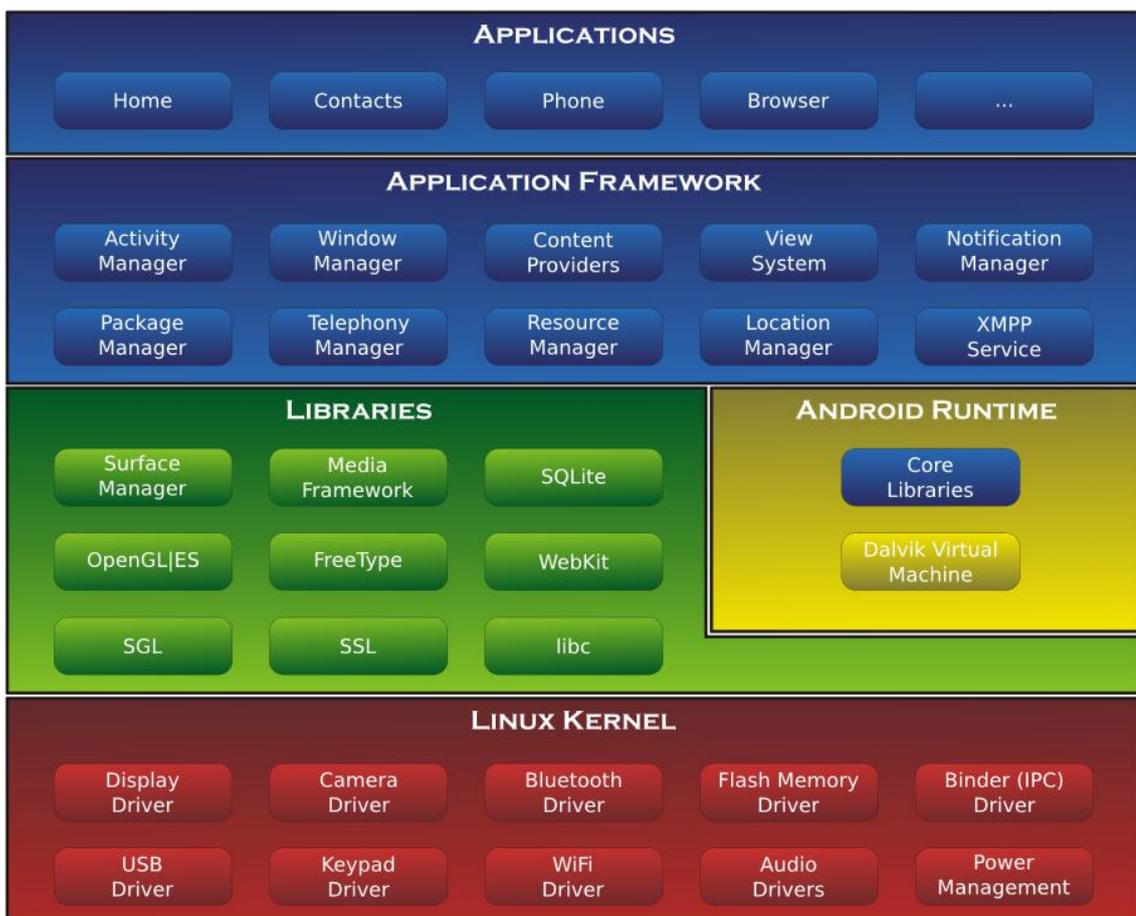


Figura 7 - Arquitetura Android (Fonte: [10])

O Google associa o sistema Android como uma pilha de softwares, onde em cada camada da pilha ficam agrupados os vários programas com funções específicas. O Kernel do Android foi criado utilizando a versão 2.6 do Linux, que inclui programas de gerenciamento de memória, software de gerenciamento de energia e drivers de gerenciamento dos recursos como câmera, Bluetooth, entre outros.

No nível acima ficam as bibliotecas e no mesmo nível desta camada, existe a máquina virtual Dalvik e um conjunto de bibliotecas do núcleo Java.

A camada de framework apresenta os programas que gerenciam as funções básicas do aparelho, alocação de recursos, gerenciamento de memória, e a mudança entre processos. Os desenvolvedores utilizam as opções disponíveis nessa camada para construir os aplicativos e aproveitar os recursos disponíveis no sistema.

Já na camada mais alta da pilha estão as aplicações e as funções básicas do aparelho. É onde os usuários tem a maior parte da interação com o dispositivo.

4.5 Offloading

Offloading ou *cyber foraging* são termos definidos por Eduardo Cuervo em 2010 e Mahadev Satyanarayanan em 2011 respectivamente, que definem a estratégia de execução de tarefas computacionais em servidores remotos com melhores capacidades computacionais e de armazenamento utilizados para aumentar o desempenho do dispositivo móvel e reduzir o consumo de energia. A execução remota de tarefas pode ser realizada em máquinas virtuais de uma nuvem pública ou em qualquer máquina da mesma rede local na qual os dispositivos móveis estão conectados, sendo este último denominado de *cloudlet*.

Cloudlet é um conceito introduzido em 2011 por Mahadev Satyanarayanan e tem como objetivo realizar o *offloading* em máquinas que estejam na mesma rede local que os dispositivos móveis utilizando as redes WiFi, que geralmente possuem velocidades maiores e latências menores. Com isso, é possível obter-se um serviço de melhor qualidade das aplicações que são sensíveis à percepção da latência.

5 A ferramenta

5.1 Filtro de acessibilidade

O filtro de acessibilidade tem como objetivo ajudar os daltônicos na identificação de contrastes não visíveis em imagens. É baseado na ideia de que os contrastes entre tons de vermelho e verde são difíceis de serem identificados por protanomalias e deuteranomalias, e que os contrastes verde-azul e vermelho-azul são notados mais facilmente.

Desta forma, o filtro foi desenvolvido para intensificar a componente azul de cada pixel da imagem através de um parâmetro i . Aumentando a intensidade do azul nos pontos onde temos alta intensidade do verde, transformamos o contraste vermelho-verde em vermelho-azul. Esta mesma técnica aplicada nas altas intensidades do vermelho, transforma o contraste vermelho-verde em verde-azul.

O valor do parâmetro de intensidade i pertence à $[-1, 1]$ e é pré-configurado conforme o tipo de daltonismo identificado através do teste de Ishihara ou através de configuração manual.

Algoritmo do filtro de acessibilidade:

$$\alpha = \begin{cases} 0 & \text{se } i \geq 0 \\ -i, & \text{caso contrário} \end{cases}$$

$$\beta = \begin{cases} 0 & \text{se } i \leq 0 \\ i, & \text{caso contrário} \end{cases}$$

$$\gamma = 1 - (\alpha + \beta)$$

Entrada: *Image img, α, β, γ*

pixels \leftarrow *img.pixels()*

Para cada p de *pixels* faça:

$$p_{\text{azul}} \leftarrow (p_{\text{azul}} * \gamma) + (p_{\text{vermelho}} * \alpha) + (p_{\text{verde}} * \beta)$$

Fim

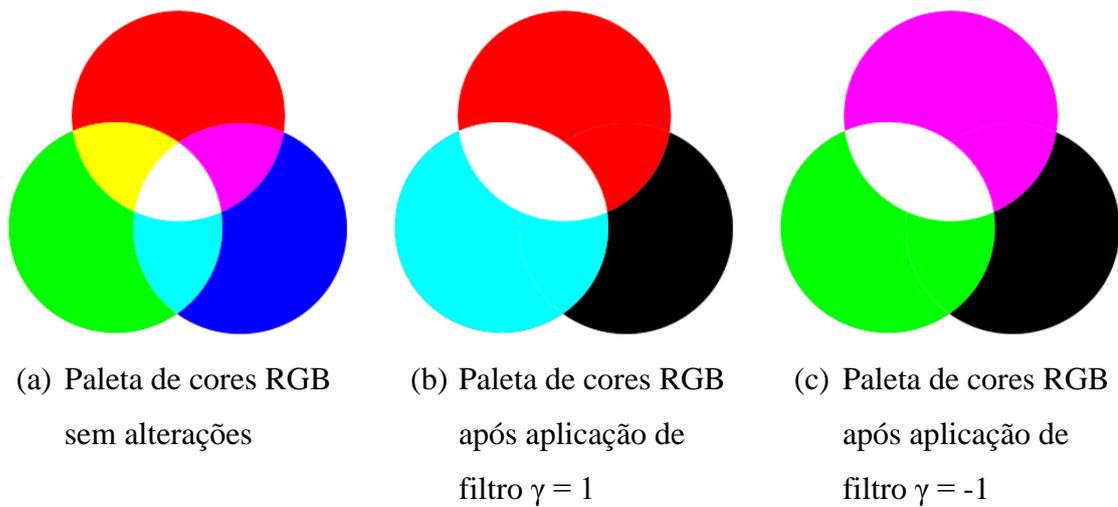


Figura 8 - Paleta de cores após aplicação de filtro de acessibilidade

5.2 Análise para *Offloading*

A análise para *offloading* tem como objetivo verificar, através de informações capturadas da rede móvel e do dispositivo, o melhor local para se realizar o processamento da imagem. Esta análise leva em consideração os recursos internos do dispositivo como memória disponível e conexão com a rede móvel, tempo de processamento local, e tempo de processamento remoto juntamente com o tempo de transmissão do arquivo.

A seguir, detalharemos as análises realizadas dos fatores que influenciam na decisão do processamento.

5.2.1 Estimativa do tempo de processamento da imagem

Um dos fatores para decidir quando realizar *offloading* é a estimativa do tempo de aplicação do filtro de acessibilidade na imagem selecionada pelo usuário no servidor e no dispositivo móvel.

Para estimar o tempo de processamento da imagem no servidor foram realizados testes da aplicação do filtro em quatorze imagens de tamanhos diferentes. O filtro foi aplicado cinco

vezes em cada imagem e o resultado do tempo de processamento foi registrado conforme tabela a seguir.

O servidor utilizado no teste possuía processador Intel®Core™2Duo CPU T6600 @ 2.20GHz com 4Gb de memória RAM, e sistema operacional Windows 7 64 bits.

#	Altura	Largura	Altura x Largura (px)	Tempo (ms)
1	233	233	54289	109
1	233	233	54289	62
1	233	233	54289	15
1	233	233	54289	16
1	233	233	54289	16
2	466	466	217156	156
2	466	466	217156	94
2	466	466	217156	31
2	466	466	217156	31
2	466	466	217156	32
3	699	699	488601	172
3	699	699	488601	109
3	699	699	488601	78
3	699	699	488601	62
3	699	699	488601	63
4	932	932	868624	250
4	932	932	868624	156
4	932	932	868624	125
4	932	932	868624	114
4	932	932	868624	109
5	1165	1165	1357225	312
5	1165	1165	1357225	218
5	1165	1165	1357225	187
5	1165	1165	1357225	156
5	1165	1165	1357225	171
6	1398	1398	1954404	359
6	1398	1398	1954404	296
6	1398	1398	1954404	250
6	1398	1398	1954404	249
6	1398	1398	1954404	234
7	1631	1631	2660161	484
7	1631	1631	2660161	406
7	1631	1631	2660161	343
7	1631	1631	2660161	344

7	1631	1631	2660161	327
8	1864	1864	3474496	546
8	1864	1864	3474496	484
8	1864	1864	3474496	468
8	1864	1864	3474496	421
8	1864	1864	3474496	437
9	2097	2097	4397409	655
9	2097	2097	4397409	608
9	2097	2097	4397409	546
9	2097	2097	4397409	532
9	2097	2097	4397409	593
10	2330	2330	5428900	813
10	2330	2330	5428900	717
10	2330	2330	5428900	670
10	2330	2330	5428900	796
10	2330	2330	5428900	688
11	2563	2563	6568969	923
11	2563	2563	6568969	905
11	2563	2563	6568969	920
11	2563	2563	6568969	812
11	2563	2563	6568969	812
12	2796	2796	7817616	1108
12	2796	2796	7817616	1014
12	2796	2796	7817616	1014
12	2796	2796	7817616	936
12	2796	2796	7817616	998
13	3029	3029	9174841	1280
13	3029	3029	9174841	1201
13	3029	3029	9174841	1170
13	3029	3029	9174841	1342
13	3029	3029	9174841	1092
14	3262	3262	10640644	1419
14	3262	3262	10640644	1373
14	3262	3262	10640644	1280
14	3262	3262	10640644	1342
14	3262	3262	10640644	1295

Tabela 2 - Tempo de processamento da imagem no servidor

A partir dos dados acima, foi gerado o gráfico de dispersão da quantidade total de pixels da imagem e tempo de execução da aplicação do filtro.

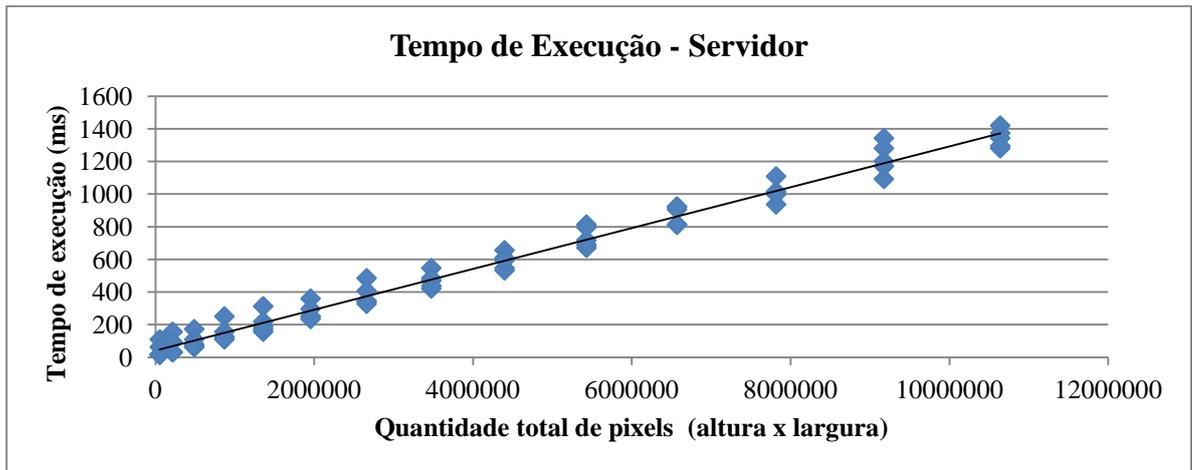


Figura 9 - Gráfico do tempo de execução x quantidade de pixels da imagem no servidor

O gráfico acima sugere um modelo linear $E(Y|x) = \alpha + \beta x$, ou seja, o tempo médio da aplicação do filtro na imagem é uma função linear da quantidade de pixels.

Para encontrar os valores de α e β do modelo linear e assim determinar a função de estimativa do tempo de execução, foram calculados os estimadores de mínimos quadrados.

Da tabela 2, obtemos as informações:

$$n = 70, \sum y_i = 37346, \sum x_i = 275516675, \sum x_i y_i = 246832376093,$$

$$\bar{y} = 533,51, \bar{x} = 3935952,5, \sum x_i^2 = 1881656615949630$$

Com isso,

$$\hat{\beta} = \frac{246832376093 - (70)(3935952,5)(533,51)}{1881656615949630 - (70)(3935952,5)^2} = 0,00013$$

$$\hat{\alpha} = 533,51 - (0,00013)(3935952,5) = 40,60300$$

O que nos dá o modelo ajustado:

$$\hat{y}_i = 40,60300 + 0,00013x_i$$

O mesmo procedimento foi realizado no dispositivo móvel para determinar sua função de estimativa do tempo de processamento. Porém, como sua memória é limitada não foi possível processar todas as imagens utilizadas nos testes do servidor.

O dispositivo móvel utilizado no teste foi um Samsung Galaxy S4 com 16 Gb de memória interna e 2 Gb de memória RAM, CPU Quad-core 1.9 GHz Krait 300, Android OS v4.4.2.

A tabela abaixo mostra o resultado dos testes:

#	Altura	Largura	Altura x Largura (px)	Tempo (ms)
1	233	233	54289	30
1	233	233	54289	58
1	233	233	54289	11
1	233	233	54289	29
1	233	233	54289	9
2	466	466	217156	111
2	466	466	217156	66
2	466	466	217156	46
2	466	466	217156	50
2	466	466	217156	71
3	699	699	488601	161
3	699	699	488601	148
3	699	699	488601	149
3	699	699	488601	164
3	699	699	488601	181
4	932	932	868624	298
4	932	932	868624	279
4	932	932	868624	296
4	932	932	868624	269
4	932	932	868624	418
5	1165	1165	1357225	557
5	1165	1165	1357225	416
5	1165	1165	1357225	507
5	1165	1165	1357225	388
5	1165	1165	1357225	369
6	1398	1398	1954404	523
6	1398	1398	1954404	477
6	1398	1398	1954404	443
6	1398	1398	1954404	548
6	1398	1398	1954404	487
7	1631	1631	2660161	780

7	1631	1631	2660161	731
7	1631	1631	2660161	719
7	1631	1631	2660161	687
7	1631	1631	2660161	663

Tabela 3 - Tempo de processamento da imagem no dispositivo móvel

A partir dos dados da tabela acima, foi gerado o gráfico de dispersão da quantidade total de pixels e tempo de execução da aplicação do filtro na imagem.

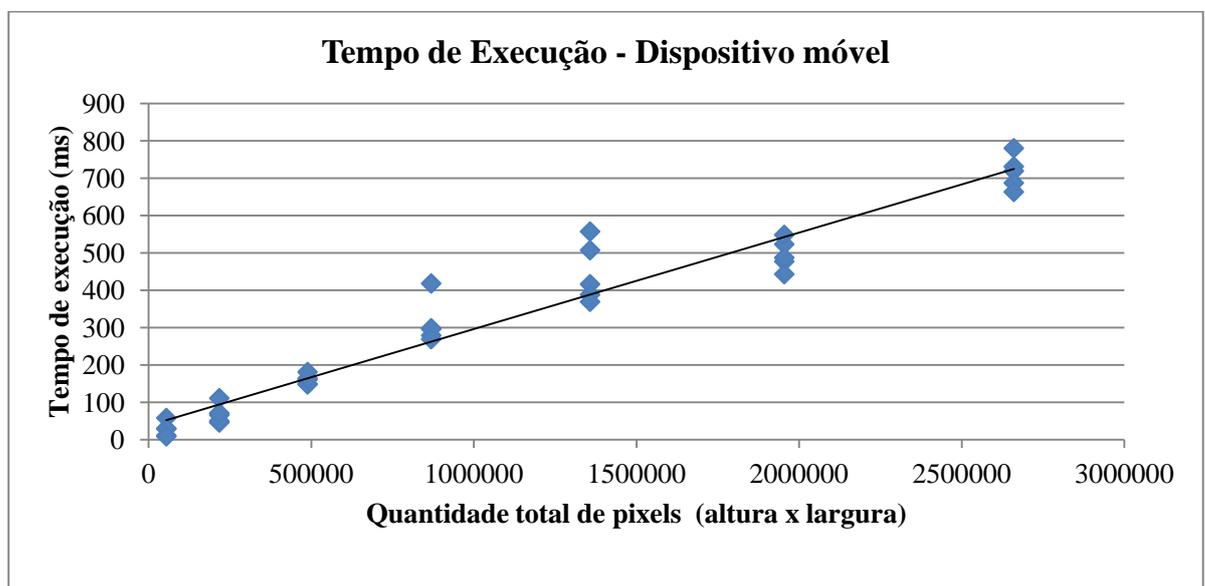


Figura 10 - Gráfico do tempo de execução x quantidade de pixels da imagem no dispositivo móvel

O gráfico acima também sugere um modelo linear $E(Y|x) = \alpha + \beta x$, ou seja, o tempo médio da aplicação do filtro na imagem é uma função linear da quantidade de pixels.

Para encontrar os valores de α e β do modelo linear e assim determinar a função de estimativa do tempo de execução, foram calculados os estimadores de mínimos quadrados.

Da tabela 3, obtemos as informações:

$$n = 35, \sum y_i = 11139, \sum x_i = 38002300, \sum x_i y_i = 19232041117,$$

$$\bar{y} = 318,26, \bar{x} = 1085780, \sum x_i^2 = 68907769280980$$

Com isso,

$$\hat{\beta} = \frac{19232041117 - (35)(1085780)(318,26)}{68907769280980 - (35)(1085780)^2} = 0,00026$$

$$\hat{\alpha} = 318,26 - (0,00026)(1085780) = 37,93092$$

O que nos dá o modelo ajustado:

$$\hat{y}_i = 37,93092 + 0,00026x_i$$

Logo, com as funções determinadas anteriormente, podemos estimar o tempo de processamento da imagem no servidor e no dispositivo móvel. Na ferramenta, este processo para identificar a função de estimativa foi automatizado, possibilitando realizar a estimativa em diversos aparelhos.

5.2.2 Estimativa do tempo de transmissão da imagem

Além de estimar o tempo de processamento da imagem no servidor e no dispositivo móvel, é necessário levar em consideração e avaliar o tempo estimado do envio e recebimento da imagem ao servidor. Porém, o tempo de transmissão pode variar conforme o tipo de conexão estabelecida com a rede móvel.

Para estimar o tempo de transmissão da imagem é necessário verificar a velocidade das redes móveis disponíveis. Então, foi desenvolvido um servidor para auxiliar neste processo. Primeiramente, um comando (recebimento ou envio de dados) é recebido para identificar qual estimativa será realizada. Caso o comando seja de estimativa do tempo de recebimento, o servidor envia a quantidade de bytes que serão transferidos, e em seguida envia o arquivo em blocos de 32 bytes para o cliente. Caso o comando seja de estimativa do tempo de envio, o servidor recebe a quantidade de bytes a serem enviados, lê e aguarda a finalização do envio dos dados.

O cliente envia o comando para iniciar o teste de recebimento ou envio de dados, e executa o procedimento. Caso o comando enviado seja o de recebimento de dados, o cliente recebe a quantidade de bytes que serão transferidos, lê os dados enviados pelo servidor, e contabiliza o tempo gasto no processo. Caso o comando seja o de envio de dados, o cliente envia a quantidade de bytes a serem transferidos, envia o arquivo em blocos de 32 bytes para o servidor, e contabiliza o tempo gasto na transferência.

Após o procedimento, as taxas de recebimento e de envio de dados são calculadas e armazenadas na aplicação. Para estimar o tempo de transferência da imagem, basta a ferramenta verificar o tamanho do arquivo a ser processado e dividir pela taxa de transmissão, obtendo o tempo estimado.

5.2.3 Relação memória disponível x memória a ser alocada

Outro fator determinante para a decisão de quando realizar *offloading* é a relação da quantidade de memória disponível no dispositivo móvel e da quantidade de memória necessária para a aplicação do filtro na imagem.

O sistema operacional Android possui uma funcionalidade nativa onde é possível identificar a quantidade de memória disponibilizada para o processo atual, e utilizando outros recursos de gerenciamento de memória da própria plataforma é possível identificar a quantidade de memória já utilizada pela aplicação, e a quantidade de memória ainda disponível.

Analisando o código desenvolvido para o processamento da imagem, algoritmo abaixo, é possível identificar a quantidade de memória que será utilizada para a execução local. Primeiramente, é alocada na memória a imagem que será enviada ao método de aplicação do filtro, quantidade total de $[largura * altura] * 4$ bytes. No método, é alocado um vetor de inteiros de tamanho $[largura * altura da imagem] * 4$ bytes para a manipulação dos pixels, e no final do processo é criada uma nova imagem, resultado da aplicação do filtro, $[largura * altura] * 4$ bytes. Portanto, a quantidade de memória a ser alocada no processamento local é de aproximadamente $3 * [largura * altura] * 4$ bytes.

Algoritmo da aplicação do filtro de acessibilidade:

Entrada: *Image img, α, β, γ*
pixels \leftarrow *img.pixels()*
Para cada *p* **de** *pixels* **faça:**
 $p_{\text{azul}} \leftarrow (p_{\text{azul}} * \gamma) + (p_{\text{vermelho}} * \alpha) + (p_{\text{verde}} * \beta)$
Fim
imgFiltered \leftarrow *img.copy(img.config)*
imgFiltered.pixels() \leftarrow *pixels*
Saída: *imgFiltered*

6 Conclusão

6.1 Resultados

A ferramenta desenvolvida para a plataforma Android permite ao usuário realizar o teste de Ishihara para identificar uma possível deficiência na percepção de cores. Além disto, permite a percepção do contraste entre tons vermelho e verde geralmente despercebido parcialmente ou por completo em pessoas protanomalias ou deuteranomalias.

A tela inicial da aplicação apresenta duas opções ao usuário (*Figura 11*). A primeira opção leva o usuário ao Menu Principal da ferramenta, onde são disponibilizadas as principais funcionalidades da aplicação: Teste de Ishihara, Acessibilidade e Configurações (*Figura 12*). A segunda opção inicia o Teste de Ishihara (*Figura 13a*), e em segundo plano inicia a operação para identificar a função de estimativa do tempo de processamento local, e a operação para identificar as taxas de transferência das redes móveis disponíveis no momento.

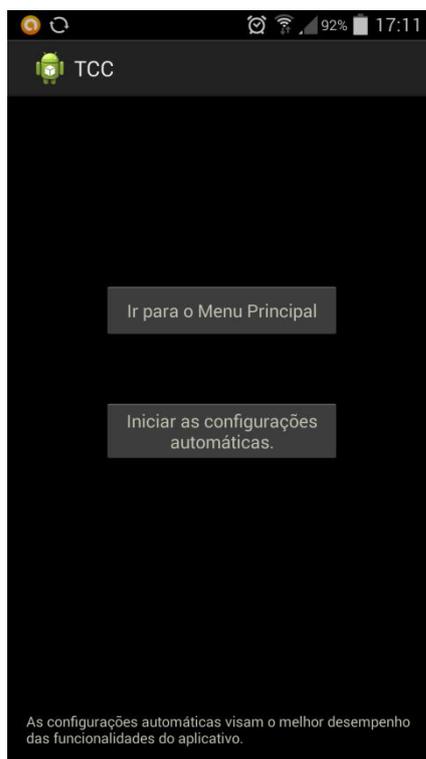


Figura 11 - Tela Inicial

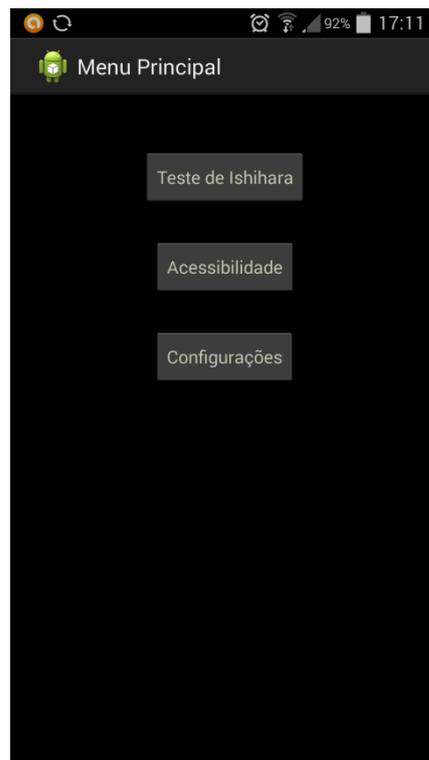
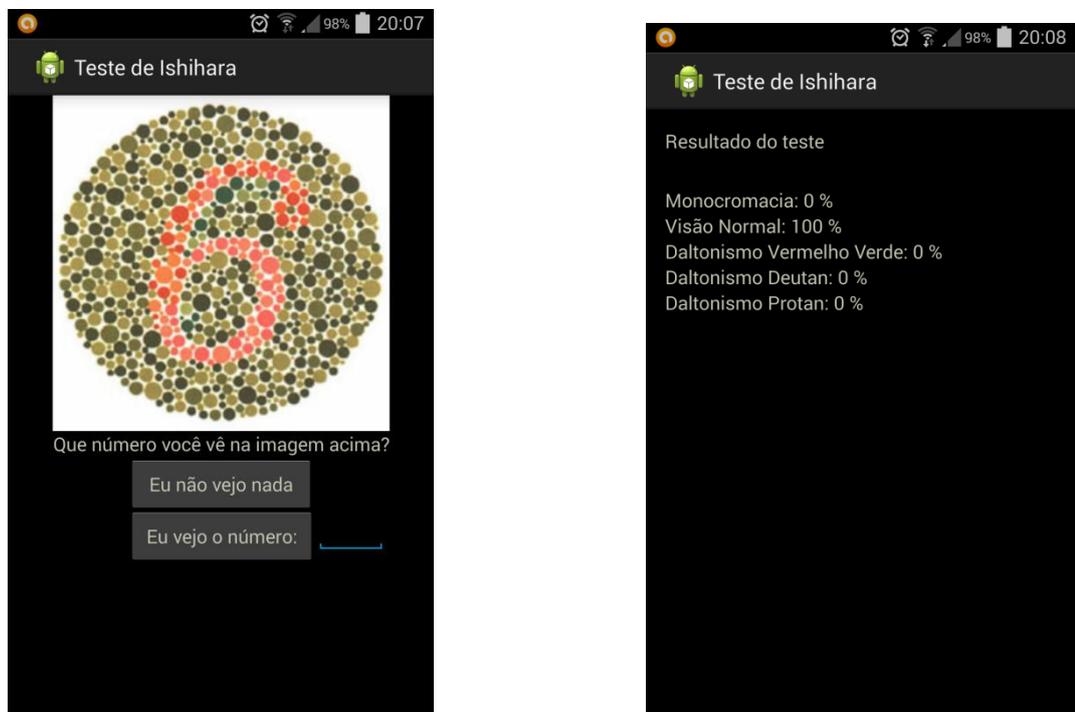


Figura 12 - Menu Principal



(a) Execução do teste de Ishihara

(b) Resultado do teste de Ishihara

Figura 13 - Teste de Ishihara

O resultado do Teste de Ishihara (*Figura 13b*) mostra a porcentagem de acertos do usuário com relação às respostas de pessoas com visão normal, com monocromacia, e com protanomia ou deuteranomalia.

Após a finalização do teste de Ishihara, e das atividades que estavam sendo executadas em segundo plano, as informações são armazenadas e exibidas no menu de configurações (*Figura 14*). Todos os campos podem ser editados manualmente. É possível ajustar a intensidade do filtro e pré-visualizar o resultado na imagem ao lado da barra de ajuste. Estão disponíveis algumas imagens utilizadas do teste de Ishihara. Caso o usuário queira verificar o resultado em outra prancha do teste, basta o mesmo clicar em cima da imagem.

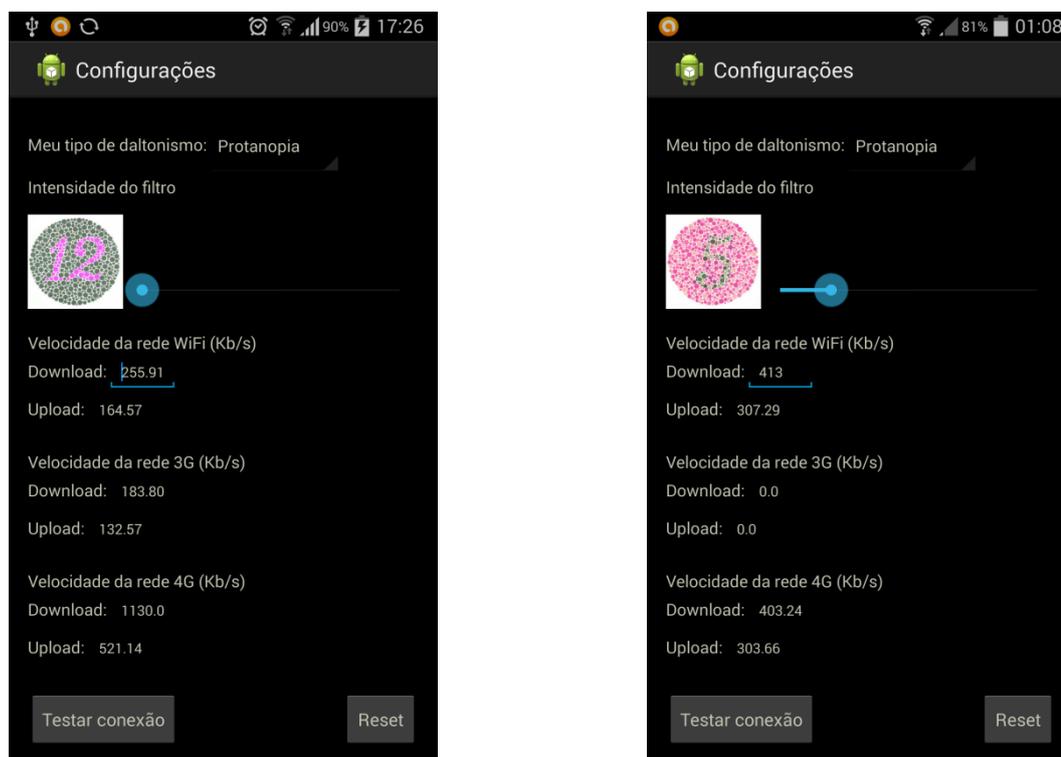
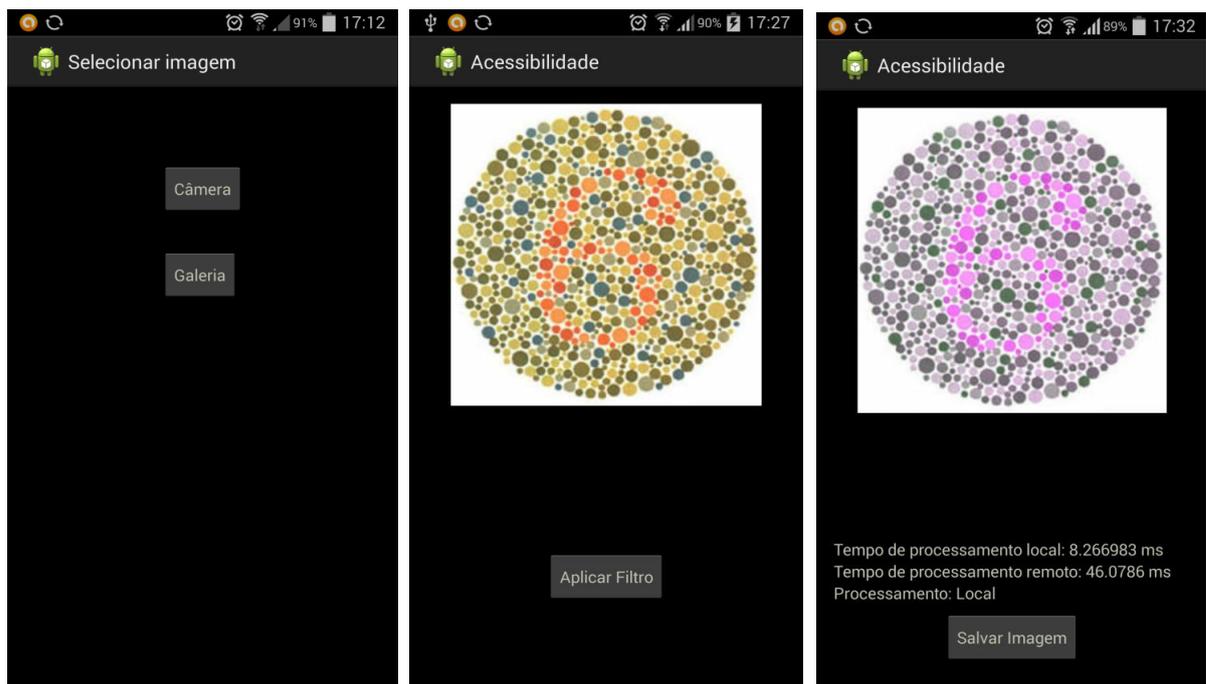


Figura 14 - Menu de configurações

A funcionalidade Acessibilidade permite ao usuário selecionar uma imagem armazenada no dispositivo móvel, ou tirar uma foto e utilizá-la como imagem de entrada (*Figura 15a*). Após a escolha, a imagem é carregada na tela, e um botão para a aplicação do filtro é apresentado.

Ao clicar no botão para aplicar o filtro (*Figura 15b*), a ferramenta verifica o melhor local para a execução, e realiza o processamento da imagem. O resultado é apresentado na tela juntamente com algumas informações, como local do processamento, e os tempos estimados para a aplicação do filtro no dispositivo móvel e no servidor (*Figura 15c*). Caso o usuário queira salvar a imagem, basta o mesmo clicar no botão Salvar Imagem.



(a) *Acessibilidade –
Seleção de imagem*

(b) *Acessibilidade –
Imagem selecionada*

(c) *Acessibilidade –
Imagem processada*

Figura 15 – Função acessibilidade

Para verificar o resultado da análise de quando realizar *offloading* foi realizado uma bateria de testes com quinze imagens de tamanhos e dimensões diferentes. O dispositivo móvel utilizado foi um Samsung Galaxy S4 com 16 Gb de memória interna e 2 Gb de memória RAM, CPU Quad-core 1.9 GHz Krait 300, Android OS v4.4.2. A aplicação do servidor desenvolvida estava rodando nas máquinas virtuais da Amazon EC2.

#	Tamanho (Kb)	Altura (px)	Largura (px)
1	31	233	233
2	89	466	466
3	155	699	699
4	202	932	932
5	293	1165	1165
6	370	1398	1398
7	459	1631	1631
8	546	1864	1864
9	690	2097	2097
10	796	2330	2330
11	796	2330	2330
12	959	2563	2563
13	1038	2796	2796
14	1247	3029	3029
15	1319	3262	3262

Tabela 4 - Informações das imagens utilizadas nos testes de offloading

Rede móvel	Taxa recebimento (Kb/s)	Taxa envio (Kb/s)
WiFi	639	927
3G	151	110
4G	399	267

Tabela 5 - Informações obtidas das redes móveis disponíveis

Com as informações da tabela 4 e 5, e informações obtidas nas configurações automáticas da ferramenta é possível estimar os tempos de processamento local e remoto, tempos de envio e recebimento de dados, verificar a quantidade de memória disponível e a ser alocada no processamento local, e verificar qual a melhor rede móvel para ser utilizada na transmissão de dados no caso do processamento remoto.

#	Tempo processamento Local (ms)	Tempo envio estimado (ms)	Tempo recebimento estimado (ms)	Tempo processamento remoto estimado (ms)	Memória disponível (Kb)	Memória a ser alocada para processar imagem (Kb)	Rede móvel a ser utilizada
1	24	49	34	46	114300	636	-
2	47	139	96	62	112420	2544	-
3	86	243	167	89	110292	5724	-
4	141	316	218	127	105476	10179	-
5	211	459	316	176	100828	15903	-
6	297	579	399	236	93956	22902	-
7	399	718	495	307	101420	31173	-
8	516	855	589	388	79908	40716	-
9	649	1080	745	480	66848	51531	-
10	797	1245	859	583	52560	63618	WiFi
11	797	1245	859	583	59608	63618	WiFi
12	961	1500	1034	697	33900	76980	WiFi
13	1141	1624	1120	822	89360	91611	WiFi
14	1336	1950	1345	958	51244	107517	WiFi
15	1547	2063	1423	1105	77576	124695	WiFi

Tabela 6 – Dados para a análise de quando realizar offloading

Utilizando estas informações, a ferramenta verificou o melhor local para o processamento, e executou a aplicação do filtro. As informações do resultado do processamento foram registradas na tabela abaixo.

#	Local processamento	Rede móvel utilizada	Tempo processamento (ms)	Tempo Envio (ms)	Tempo Recebimento (ms)
1	Local	-	5	0	0
2	Local	-	67	0	0
3	Local	-	85	0	0
4	Local	-	123	0	0
5	Local	-	169	0	0
6	Local	-	272	0	0
7	Local	-	392	0	0
8	Local	-	393	0	0
9	Local	-	468	0	0
10	Remoto	WiFi	13611	1717	10176
11	Remoto	WiFi	3372	835	551
12	Remoto	WiFi	3115	638	588
13	Remoto	WiFi	3887	903	706
14	Remoto	WiFi	5500	1184	769
15	Remoto	WiFi	4115	626	825

Tabela 7 - Informações do processamento das imagens

Da tabela 6 é possível verificar que para as imagens dos testes de 1 a 9 havia memória disponível para o processamento, e o tempo estimado de processamento local era menor do que o tempo estimado do processamento remoto. E da tabela 7, verificamos que a aplicação do filtro foi realizada no próprio dispositivo.

Para as imagens dos testes de 10 a 15, verificamos da tabela 6 que a quantidade de memória a ser alocada no processamento local seria maior do que a quantidade disponível. Da tabela 5, é possível notar que a rede móvel com as melhores taxas de transferência era a rede WiFi, o que explica a decisão da ferramenta em realizar *offloading*.

Esta técnica mostrou-se uma boa alternativa para resolver problemas do desempenho limitado dos dispositivos móveis devido aos recursos disponíveis. No entanto, se a conexão com a rede móvel do dispositivo não for boa, o consumo de energia pode ser elevado, pois o tempo de transferência do arquivo e o tempo de espera do processamento podem ser altos.

Por outro lado, as empresas de telecomunicações estão investindo cada vez mais na infraestrutura de rede, instalando novas antenas 3G e 4G, e instalando pontos de rede WiFi.

Com este último, pode-se diminuir o tráfego de dados em antenas 3G e 4G, conseqüentemente diminuindo também a latência. Desta forma, a técnica de *offloading* e seu desempenho podem melhorar.

6.2 Trabalhos Futuros

Uma possibilidade de continuação do trabalho seria incluir no estudo o fator energia utilizada no processamento da imagem. Seria interessante estudar e comparar a quantidade de energia utilizada no processamento local, e energia utilizada no processamento remoto, e considerar esta relação no momento de decisão da utilização do *offloading*.

O trabalho atual teve foco no desenvolvimento de uma ferramenta de acessibilidade voltada para a protanomalia e deuteranomalia. Para outras anomalias, é necessária a atualização do filtro, alteração que inclusive foi mencionada no trabalho anterior em que esse trabalho se baseou. Desta forma a ferramenta cobriria todos os tipos de deficiência em cores.

Outra possibilidade seria o desenvolvimento da ferramenta para as plataformas iOS da Apple e Windows Phone da Microsoft, expandindo a quantidade de usuários, e atingindo a grande maioria do mercado de dispositivos móveis, não restringindo à apenas usuários da plataforma Android.

7 Referência bibliográfica

- [1] Color Perception. <http://webvision.med.utah.edu/book/part-viii-gabac-receptors/color-perception/> - acessado em junho de 2014.
- [2] A. Asato e R. Gonçalves. Visocor - Sistema de Acessibilidade Visual. Instituto de Matemática e Estatística - Universidade de São Paulo, 2009.
- [3] P. Costa, P. Rego, E. Coutinho, F. Trinta, J. Souza. Uma Análise do Impacto da Qualidade da Internet Móvel na Utilização de Cloudlets, Universidade Federal do Ceará , 2014.
- [4] Vischeck. <http://www.vischeck.com/> - acessado em abril de 2014.
- [5] HueVue. <https://itunes.apple.com/us/app/huevue-colorblind-tools/id318177578?mt=8> - acessado abril de 2014.
- [6] Chromatic Vision Simulator. <http://asada.tukusi.ne.jp/cvsimulator/e/index.html/> - acessado em abril de 2014.
- [7] ColorBlind Vision. <http://www.givewaygames.com/apps/colorblind-vision/> - acessado em abril de 2014.
- [8] S. Ishihara. Tests for Colour-Blindness, 1972.
- [9] Android Developer Website. <http://developer.android.com/> - acessado em abril de 2014.
- [10] The free encyclopedia wikipedia. <http://en.wikipedia.org/> - acessado em setembro de 2014.
- [11] A enciclopédia livre wikipedia. <http://pt.wikipedia.org/> - acessado em setembro de 2014.
- [12] <http://www.colorado.edu/ASEN/asen5519/17light.htm/> - acessado em julho de 2014.
- [13] <http://www.gentequeeduca.org.br/planos-de-aula/de-olho-no-olho/> - acessado em julho de 2014.
- [14] J. Amabis and G. Martho. Biologia dos Organismos: A diversidade dos seres vivos. São Paulo: Moderna, 2004.

- [15] S. Merin. *Inherited Eye Diseases: Diagnosis and Clinical Management*. CRC Press, 2013.

Parte Subjetiva

1 Desafios e frustrações

Apesar de restar apenas o trabalho de conclusão de curso para finalizar a minha graduação, um dos meus maiores desafios foi lidar com o tempo. Tive muitos problemas em consolidar o tempo para realizar os estudos, pesquisas e desenvolvimento da aplicação com o trabalho na empresa que estou atualmente. Isto resultou em realizar todos os estudos e atividades da disciplina nas madrugadas dos dias de semana e também nos finais de semana, o que foi bastante cansativo.

Encontrei muita dificuldade no início do desenvolvimento da ferramenta, pois até então não tinha muito contato com a plataforma Android, apesar de conhecer a linguagem de programação utilizada. Em diversos momentos tive dificuldades em resolver alguns *bugs* encontrados no desenvolvimento, o que consumia um bom tempo até a sua resolução.

No meio do projeto, pesquisei mais a fundo sobre filtros de acessibilidade para os daltônicos do tipo tritanomalia, e iniciei o desenvolvimento de um filtro, porém devido à escassez de tempo optei por parar a implementação para dar foco nos estudos sobre *offloading* e na escrita da monografia. Este foi um item que gostaria de ter concluído e adicionado à ferramenta para que no final tivéssemos uma aplicação que cobrisse todos os tipos de daltonismo.

2 Disciplinas mais relevantes para o trabalho

MAC0110 – Introdução à Computação

MAC0122 – Princípios de Desenvolvimento de Algoritmos

MAC0323 – Estrutura de Dados

Estas disciplinas foram fundamentais para o aprendizado ao longo do curso de computação. Através delas, aprendemos os fundamentos de programação, estudamos a eficiência e a implementação de algoritmos, além das diversas estruturas de dados e suas aplicações.

MAC0211 – Laboratório de Programação I

MAC0242 – Laboratório de Programação II

Disciplinas importantes, onde foi possível desenvolver projetos de grande porte.

MAC0332 – Engenharia de Software

A disciplina engenharia de software foi importante para aprendermos sobre o gerenciamento de projetos. Através dela, aprendemos as atividades que compõe um projeto, o gerenciamento de cada tarefa e o processo de desenvolvimento de um software.

MAE0121 – Introdução a Probabilidade e a Estatística I

MAE0212 – Introdução à Probabilidade e à Estatística II

As matérias de estatística foram relevantes para a determinação das funções de estimativa dos tempos de processamento nos dispositivos móveis e no servidor.