

Say2me - Um Modelo de Reconhecimento de Imagens para Deficientes Visuais

Tonismar Régis Bernardo¹, Cristiano André da Costa¹

¹ Especialização em Desenvolvimento de Aplicações para Dispositivos Móveis - UNISINOS.

tonismar.at.gmail.com

Abstract. *This article introduces the development of the Say2me application which, through an informed image, queries a remote image description service and returns its description. The objective of this work is, using the model, to approximate visually impaired users to the social interaction of people in general. The application allows an image to be photographed or an image already in the gallery is retrieved and sent to be described. The prototype can be used on devices using the iOS operating system and was implemented using only the Swift language. In the tests performed, the model answered the description requests, with a high assertiveness index, in the average of thirty-three seconds, a time considered acceptable by the user. Although some improvements are considered, the application reaches the scope proposed by this work.*

Resumo. *Este artigo apresenta o desenvolvimento do aplicativo Say2me que, através de uma imagem informada, efetua uma consulta em um serviço remoto de descrição de imagem e retorna sua descrição. O objetivo deste trabalho é, através do modelo, aproximar usuários com deficiência visual ao convívio social das pessoas de uma forma geral. O aplicativo permite que uma imagem seja fotografada ou que seja recuperada uma imagem já existente na galeria e assim enviada para ser descrita. O protótipo pode ser utilizado em dispositivos utilizando o sistema operacional iOS e foi implementado utilizando apenas linguagem Swift. Nos testes efetuados o modelo respondeu as requisições de descrição, com alto índice de assertividade, na média de trinta e três segundos, um tempo considerado aceitável pelo usuário. Apesar de algumas melhorias serem consideradas, o aplicativo atinge o escopo proposto pelo presente trabalho.*

1. Introdução

Atualmente há um grande número de pesquisas voltadas para os diversos tipos de pessoas portadoras de necessidades especiais. Essas pesquisas visam trazer tais pessoas para o convívio social como qualquer outro cidadão. No Brasil 25 milhões (14% da população) de brasileiros possuem algum tipo de deficiência e desses, 40% possui deficiência visual parcial ou total (1).

Em Janeiro/2017 existiam 2,2 milhões de aplicativos disponíveis na *Apple Store* (2) e 2,8 milhões na *Google Play* em Março/2017 (3). Diferente da grande quantidade de aplicativos disponíveis nas duas lojas para todas as pessoas, a quantidade deles voltado para o público com necessidades especiais é bem reduzido.

Inúmeras empresas vêm desenvolvendo aplicativos que utilizam alguma forma de inteligência artificial. Podemos citar como exemplo os *bots* em aplicativos de bate-papo, a Alexa software dos dispositivos da Amazon e a Siri software dos dispositivos Apple. Muitos desses softwares possuem uma *interface* de programação para aplicativos (API) que disponibiliza o acesso ao seu conhecimento por outros softwares. Baseados em redes neurais, quanto mais utilizada suas inteligências artificiais forem, mais conhecimento será adquirido, auxiliando no aumento do percentual de acerto de suas respostas. Existem diversos tipos de APIs cada uma com um domínio específico de conhecimento. Apesar desses aplicativos serem de grande auxílio para seus usuários, até o momento da concepção deste artigo, nenhum dos exemplos citados focou sua inteligência no reconhecimento de imagens. Isso comprova que o campo de pesquisas em acessibilidade para pessoas com pouca ou nenhuma visão utilizando reconhecimento de imagens pode ser mais explorado.

Este modelo demanda os princípios da Tecnologia Assistiva (4) que é "uma área do conhecimento, de característica interdisciplinar que engloba produtos, recursos, metodologias, estratégias, práticas e serviços que objetivam promover a funcionalidade, relacionada à atividade e participação de pessoas com deficiências, incapacidades ou mobilidade reduzida, na busca de sua autonomia, independência, qualidade de vida e inclusão social". Com os avanços de *softwares* e *hardwares* nos dispositivos móveis como *smartphones* e *tablets* esses tornam-se grandes aliados para alcançar os benefícios da tecnologia assistiva.

1.1. Aplicativos Existentes

Na busca de alguns exemplos de aplicativos voltados para pessoas com problemas visuais grave ou sem nenhuma visão, podemos destacar os três seguintes exemplos:

- *BlindTool* - apenas disponível na plataforma Android, foi desenvolvido na Universidade de Massachusetts pelo cientista de computação Joseph Cohen. O aplicativo reconhece objetos. Basta o usuário apontar a câmera do celular para a sua volta e, no momento que senti-lo vibrar, significa que o aplicativo reconheceu algum objeto. A leitura tridimensional é feita por uma rede neural que relaciona o que está sendo visualizado pela câmera com um banco de imagens previamente armazenadas. O aplicativo é gratuito.(5)
- *Be My Eyes* - funciona como uma rede social, você pode se cadastrar como um voluntário ou como quem precisa de ajuda. O deficiente visual tem toda a acessibilidade necessária para solicitar uma ajuda. Assim que um voluntário recebe a solicitação ele envia as orientações que o aplicativo "lê" para o deficiente visual. Disponível gratuitamente apenas para iOS.(6)
- *Aipoly Vison* - é, praticamente, a versão para iOS do *BlindTool*, possui as mesmas características que o aplicativo para Android, também disponível gratuitamente.(7)
- *Identifi* - aplicativo para iOS bem recente, com menos de um ano disponível na *App Store*, é o que mais se assemelha ao modelo pois também faz a consulta de descrição em API remota. Apesar do aplicativo ter a opção de descrever no idioma Português brasileiro, a tradução da descrição é incorreta e com sotaque estrangeiro, confundindo o usuário. Se em algum momento o usuário não entender a descrição e quiser ouvi-la novamente, será necessário fotografar o assunto ou objeto e, novamente, enviar para o serviço de descrição e aguardar a resposta.

Já no protótipo proposto, basta o usuário clicar sobre a foto recém descrita que instantaneamente ouvirá a mensagem, sem ser necessário uma nova consulta ao serviço remoto. Além disso, o *Identifi* não salva as imagens já descritas pelo aplicativo na galeria de foto para, se necessário, posterior uso como o modelo sugerido faz.(8)

Analisando aplicativos acima citados, com exceção do *Be My Eyes*, os demais têm como característica o emprego de algum tipo de inteligência artificial (IA) para "enxergar" o que está sendo exibido diante da câmera do celular.

1.2. Trabalhos Relacionados

Trabalhos com as palavras-chave: deficiência visual, reconhecimento de imagem, dispositivos móveis, foram considerados como trabalhos relacionados.

Dentre alguns trabalhos encontrados, destacamos a pesquisa de Kelly e Maurício (9) que apresenta ideias sobre a construção de uma aplicativo para auxiliar deficientes visuais no cruzamento de vias públicas. Isso feito através de uma análise dos desafios de soluções no desenvolvimento de *interfaces* móveis que executam processamento de imagens e reconhecimento de voz na detecção de faixa de pedestres. Devido a limitação do domínio de detecção de imagem, o aplicativo não faz nenhuma consulta a serviços remotos, possuindo seu próprio algoritmo de reconhecimento. O trabalho contribui na investigação de questões presentes no universo da visão computacional para dispositivos móveis em cenários não controlados e a criação de *interfaces* para deficientes visuais.

O estudo realizado por Matusiak, Skulimowski e Strumillo (10), descreve as principais características dos módulos de software desenvolvidos para *smartphone* Android dedicados aos usuário cegos. Um módulo principal que reconhece objetos escaneados e compara-os com uma base de dados de objetos pré-existente no celular. Um módulo capaz de detectar regiões de maior brilho, ou seja, direcionar onde está a luz. Além de um outro módulo capaz de detectar a cor dos objetos. Este trabalho coopera nas pesquisas de soluções em dispositivos móveis para pessoas sem visão. O projeto teve parte financiada pelo Centro Nacional de Desenvolvimento e Pesquisa da Polônia nos anos 2010-2013.

Assim como o trabalho anterior, este último também possui um domínio de imagens a ser detectadas limitado e possui o algoritmo de detecção no próprio modelo. O trabalho foca em imagens já previamente retratadas como, por exemplo, embalagens de produtos utilizadas no dia-a-dia de um deficiente visual.

1.3. Objetivos

Inicialmente a motivação desse trabalho surgiu devido a uma conversa na qual foi demonstrado um estudo (11) onde o problema contempla a busca de uma padronização de áudioscrições para deficientes visuais. Diferente do modelo proposto, esse trabalho não utiliza nenhuma tecnologia de inteligência artificial nas descrições. As descrições são previamente gravadas por pessoas preparadas. Foi quando percebeu-se que, mesmo fora de um padrão audiodescritivo, uma pesquisa no campo da descrição automatizada de conteúdo de imagens traria benefícios para a comunidade científica voltada para a acessibilidade assistiva aos portadores de problemas visuais. Outro fato que contribuiu para a concepção desse modelo foi mobilizar o autor a sair da mesmice de cadastros, relatórios

e consultas feitas repetidamente no ambiente de trabalho e aprender novas linguagens e tecnologias.

Ao final desse estudo é pretendido um modelo funcional de um aplicativo que, através de consultas a um serviço de inteligência artificial disponível na *Internet*, descrever uma foto recém fotografada com o celular e também fotos existentes na galeria de imagens visando auxiliar portadores de deficiências visuais.

2. Fundamentação Teórica

Existem hoje disponíveis inúmeros serviços na área da inteligência artificial dos mais diversos tipos. Criados com a utilização das tecnologias de computação em nuvens, redes neurais e *deep learning*, empresas como Google, Microsoft, Amazon e IBM desenvolveram ótimos produtos baseados na sua necessidade de utilização.

Contida no campo da ciência da computação ligado a inteligência artificial, redes neurais artificiais, inspiradas nas redes neurais biológicas, são sistemas paralelamente distribuídos compostos por unidades de processamento simples: os neurônios. Tais neurônios geralmente são conectados baseados a pesos, adquiridos através de funções matemáticas, que armazenam o conhecimento. Representado por um modelo, esse conhecimento serve para ponderar a entrada recebida da rede (12). Tais funções matemáticas buscam se assemelhar as estruturas neurais biológicas, dando capacidade de se adaptar aos seus parâmetros como resultado da interação com o meio externo, melhorando gradativamente o seu desempenho na solução de um determinado problema(13).

Para tratar da dificuldade de proporcionalidade baseado no volume do espaço em que os dados são inseridos, surge o *Deep Learning*. Inspirado no estudo do córtex visual dos mamíferos a abrangência do *Deep Learning* pode ser definida como uma sub-área de Aprendizado de Máquina(12). *Deep learning* permite que modelos computacionais compostos por camadas de multiprocessamento aprendam representação de dados com múltiplos níveis de abstração. Esses métodos melhoram drasticamente o estado da arte em reconhecimento da fala, reconhecimento de objetos visuais e muitos outros domínios. Através do algoritmo de *backpropagation* que descobre a estrutura intrincada em grandes conjuntos de dados. O *Deep learning* trouxe grandes avanços no processamento de imagens, vídeo, fala e áudio (14).

O emprego das tecnologias acima citadas necessita de um grande poder de processamento, neste ponto outra tecnologia se faz necessária, a Computação em Nuvem (*Cloud Computing*). Segundo Marins(15), computação em nuvem é uma expressão genérica que descreve a evolução de tecnologias e processos, compostos de serviços, aplicações, informações e infraestrutura distribuídos, de modo que estes possam ser arranjados dinâmica, elástica e rapidamente na medida em que forem consumidos. Através da nuvem é possível terceirizar uma demanda computacional e recuperar seu resultado. Empresas de grande porte disponibilizam tal terceirização através de produtos de diferentes domínios e valores. Os produtos: Azure da Microsoft(16), Cloud Vision do Google(17), Watson da IBM(18) e CloudSight(19) são exemplos de serviços de descrição de imagens encontrados atualmente.

A disponibilização desses serviços é feita através de *web services*. Segundo W3C (20) *Web service* é um sistema de software designado para suportar interoperável interação

máquina-para-máquina através da rede. Possui uma *interface* descrita em um formato processável em máquina (especificamente WSDL). Outros sistemas interagem com o serviço da Web de uma maneira prescrita por sua descrição usando mensagens SOAP, normalmente transmitidas através do HTTP com uma serialização XML em conjunto com outros padrões relacionados à Web.

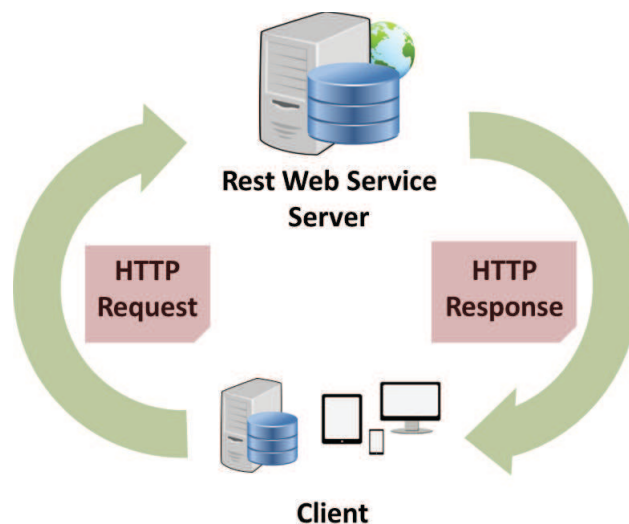


Figura 1. Arquitetura de utilização de um *web service*

A figura 1 apresenta, de uma forma simplificada, a utilização de um serviço disponível. O cliente que utiliza o serviço envia, através da rede, uma pergunta (*HTTP Request*) em um formato pré definido para o servidor (*Rest Web Service Server*), este processa a requisição e retorna a resposta (*HTTP Response*) para o aplicativo.

3. Implementação

Para a concepção do modelo foi escolhida a plataforma iOS com desenvolvimento na linguagem Swift (21). A escolha teve como base o bom aproveitamento do módulo na plataforma ao decorrer do curso, a ótima curva de aprendizagem do Swift, a familiaridade com os dispositivos da Apple e o ótimo suporte a acessibilidade no sistema operacional iOS. Além das justificativas já citadas, houve um interesse em evitar o problema da grande ramificação de versões e tamanhos de telas existente no desenvolvimento da plataforma Android.

3.1. Desenvolvimento

O protótipo foi desenvolvido utilizando o ambiente de desenvolvimento integrado (IDE) Xcode na versão 8.3.3, apenas na linguagem Swift e utiliza uma biblioteca externa (Pod) para executar as chamadas com o *Web Service*. O aplicativo possui três principais funções: fotografar e utilizar esta imagem, buscar uma imagem na biblioteca de imagens e, em quaisquer das duas opções, descrever a imagem em questão. Tanto para fotografar quanto para buscar imagens na galeria, nesta versão inicial, é utilizada a forma simples de acesso via classe *UIImagePickerController* (22) em conjunto com seu *delegate*, não foi implementado uma *interface* nativa com controles de fotografar no modelo. A figura 2 contém o diagrama de sequência para as principais funções do modelo.

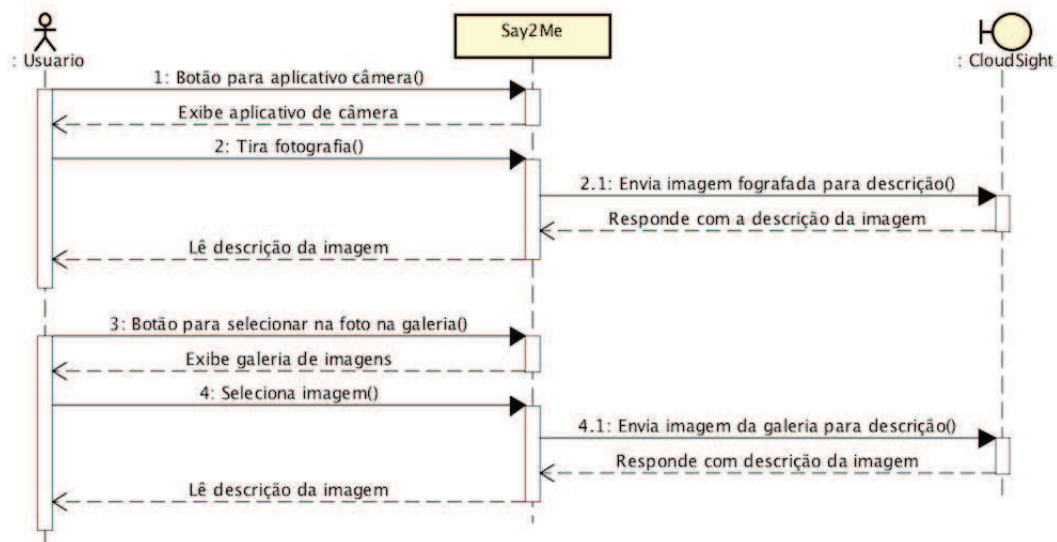


Figura 2. Diagrama de sequência para fotografar e selecionar na galeria.

3.2. Layout

O *layout* foi elaborado visando a facilidade de uso, visto que é um aplicativo com foco nos usuários sem visão. A principal atenção foi dada nas mensagens que são emitidas ao ativar o recurso de leitura de telas nativo nos dispositivos da Apple, VoiceOver (23). Cada botão possui uma descrição do que ele executa, assim como a imagem da galeria ou fotografada, também possui uma descrição ao ser selecionada.



Figura 3. Tela com aba para buscar imagem na galeria ativada.

O aplicativo, conforme figura 3, possui uma tela inicial separada em três abas através de uma barra na parte inferior com três botões: um botão para ir para a aba inicial com ícone de uma máquina fotográfica, esta aba possui o botão para abrir o aplicativo Câmera do dispositivo; um botão para selecionar a aba central, com ícone de várias fotos empilhadas, esta aba possui o botão para abrir a galeria de imagens para escolha; o terceiro

botão, com ícone de engrenagem, é exibida tela com opções. Esta última aba não foi desenvolvida e será melhor explicada no item Atividades Futuras.

3.3. Escolha da API

Para escolha do serviço de descrição de imagem foram testadas as APIs Azure(16), Cloud Vision(17), Watson(18) e CloudSight(19). A concepção do teste utilizou como base o artigo de Tyler Keenan (24). Foram enviados sete tipos diferentes de imagens para cada serviço: uma imagem de uma celebridade, uma paisagem, um objeto, um animal, uma multidão (imagem em preto e branco), um carro e uma pintura em um quadro.

Tanto Cloud Vision e Watson retornam uma lista de possíveis itens existentes na imagem elencados por um valor na forma de pontuação. Quanto maior essa pontuação maior a possibilidade deste item ser o que melhor descreve a foto. As APIs CloudSight e Azure trazem um resultado mais amigável para as necessidades do modelo porém, os resultados da CloudSight ficaram bem acima das demais.

Em adição aos resultados, a CloudSight exige menor burocracia para utilizar o serviço gratuitamente para testes. Enquanto que os três demais serviços oferecem um período de 30 dias corridos, CloudSight oferece quinhentas requisições sem um limite de período. Para ativar uma conta basta informar nome e e-mail, já os demais serviços requisitam dados de documentos pessoais e um cartão de crédito internacional.

Como foi implementado apenas um tipo dos serviços testados, não foi possível testar a velocidade de resposta de cada um. O envio das imagens para a tabela 1 foi feito através da página de demonstração de cada serviço, onde cada um utiliza sua estrutura de internet.

Nos resultados pode notar que tanto Azure quanto CloudSight reconheceram a personalidade existente na imagem, o segundo serviço ainda citou a família da celebridade. Os resultados das outras duas opções deixaram bastante a desejar. Outro ponto interessante foi a ótima descrição da paisagem feita pelo serviço da Microsoft, CloudSight apesar de descrever amigavelmente, ficou devendo comparado a descrição do Azure. Em compensação, a descrição do objeto bicicleta pelo Azure foi além do necessário, aparentemente por não haver informações de chão ou fundo na imagem, o serviço entendeu que o objeto estava no ar. O serviço repetiu o mesmo erro com a foto do cachorro, não existe o *frisbee* interpretado e também com a multidão onde foi descrita uma moto inexistente. Com a imagem do carro, apesar de todas as APIs terem identificado um veículo, CloudSight interpretou o modelo e a cor acertadamente. Assim também aconteceu com o quadro da Mona Lisa, apenas o mesmo serviço detectou a famosa pintura enquanto que os demais serviços entenderam que tratava-se de um quadro, exceto o IBM Watson.

Ao efetuar os testes percebeu-se que alguns resultados alteram se a mesma imagem é enviada mais de uma vez para reconhecimento. Como para o protótipo a imagem será enviada apenas uma única vez, para os testes foi levado em consideração apenas o resultado do primeiro envio. Os resultados podem ser analisado na tabela 1 e na tabela 2 estão expostas as imagens analisadas.

	Cloud Vision	Azure	Watson	CloudSight
Pessoas	beauty	Kobe Bryant et al. posing for a photo	person	Kobe Bryant and family
Paisagem	nature	a lake surrounded by a body of water with a mountain in the background	blue color	River beside mountain photography
Objeto	bicycle	a bicycle is jumping in the air	bicycle	stainless steel road bicycle
Animal	dog breed	a brown and white dog playing with a frisbee	dog	tricolor short-coated puppy
Multidão (P/B)	Black and white crowd	a group of people on a motorcycle in front of a crowd	black color	grayscale photo of topless man
Carro	car	a close up of a car	body (of vehicle)	yellow Lamborghini Aventador S
Quadro	picture frame	a painting in a frame hanging on a wall	olive green color	monalise painting

Tabela 1. Tabela de comparação de descrição de imagem.



Tabela 2. Relação de imagens analisadas.

3.4. CloudSight

A API CloudSight trabalha com dois passos básicos. Envia-se uma imagem através de uma requisição HTTP POST e então solicita-se uma resposta usando uma ou mais

requisições. Quando a imagem é enviada para API, se houve sucesso no recebimento, um *token* é retornado como resposta (figura 4). Utilizando esse *token*, outra requisição é feita e então a descrição da imagem enviada anteriormente é retornada (figura 5).

CloudSight tem suporte a autenticação OAuth1-Simple: onde o cliente calcula uma assinatura de uma chave, segredo, *nonce*, URL, parâmetros incluídos na requisição e inclui os dados resultantes no cabeçalho *Authorization* HTTP. Ou pode-se utilizar o método baseado em chave simples onde cada requisição é incluído um cabeçalho: *Authorization: CloudSight [chave]*.

O modelo foi implementado utilizando a biblioteca *CloudSight* (POD) disponibilizada pelos desenvolvedores da API através do *CloudSightQueryDelegate* com autenticação OAuth1-Simple. Após cadastro efetuado no site, basta acessá-lo com os dados de usuário e senha e criar um novo projeto que a chave e o segredo serão disponibilizados. Dependendo do tipo de usuário: *trial*, *small*, *medium*, *large*, é possível criar mais de um projeto com um par de chave/segredo para cada projeto.

```
Definition
http://api.cloudsight.ai/image_requests

Result Format
● 200 OK · ● 400 Bad Request
{
  "url": "//images.cloudsightapi.com/uploads/i
image_request/image/19/19404/19404152/Image.jpg
",
  "token": "AJKAWHKGLjqMd9KDNIXQfg",
}
```

Figura 4. Resposta com o token da imagem enviada.

```
Definition
http://api.cloudsight.ai/image_responses/:token

Result Format
● 200 OK · ● 200 OK · ● 200 OK · ● 404 Not Found ·
● 200 OK
{
  "status" : "completed",
  "name" : "red beats by dre headphones",
  "flags" : ["adult"],
  "ttl" : 60,
  "url" : "http://.../your_headphones_image.jp
g",
  "token" : "WySLTJWESPTt6v0o8mzKf"
}
```

Figura 5. Resposta com a descrição da imagem.

3.5. Recursos

Apesar do serviço VoiceOver ditar as ações estabelecidas no aplicativo, foi implementado também uma função utilizando a classe *AVFoundation* para que, no momento que a descrição da imagem é recebida, está já seja "falada" para o usuário sem que seja necessário uma ação, trazendo um dinamismo na utilização do modelo. Se o usuário quiser

que a descrição seja dita novamente, basta clicar na imagem. Dessa forma, é possível a utilização do aplicativo sem a função de leitura de telas ativada.

4. Testes e Desenvolvimentos Futuros

Para os testes foram utilizados um iPhone 6s Plus e um iPhone 5c, ambos com versão 10.x do iOS. Não houveram diferenças consideráveis do modelo sendo executado nos dois dispositivos. O layout se adaptou da mesma forma nos dois tamanhos de telas e não ocorreu diferença de desempenho entre os celulares. A coleta de dados foi feita através do envio de nove tipos diferentes de imagens quatro vezes seguidas para cada imagem. Foi cronometrado o tempo entre a requisição da descrição (envio da imagem) até a sua recepção, a velocidade de conexão utilizada foi de trinta *megabits* por segundos. A média de tempo de resposta de cada imagem foi de trinta e três segundos e o desvio padrão de dezesseis segundos. Como o tempo de resposta mostrou-se bastante aleatório mantendo um alto índice de assertividade, independente do tipo de imagem enviada, a média foi calculada somando todos os tempos de envio de cada imagem. A quantidade de quatro envios seguidos para cada foto foi definida em função de que, a partir do segundo envio, a descrição passa a não mais sofrer alterações. O gráfico na figura 6 demonstra a variação de tempo na respostas para cada envio.

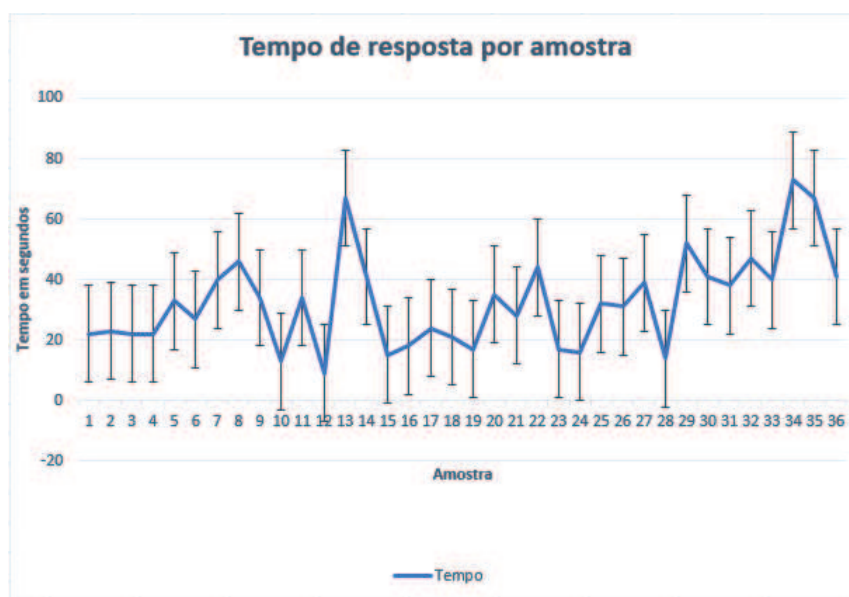


Figura 6. Gráfico da variação do tempo de envio.

Para que os testes fossem possíveis seria necessário um perfil com deficiência visual, usuário de iPhone e que entendesse inglês. Diante da ausência desse específico perfil, a utilização do aplicativo nos testes foi feita por uma pessoa simulando não enxergar (com e sem o recurso VoiceOver) constatando que o modelo contempla o escopo previsto para o trabalho. Como comprovado anteriormente, a média de 33 segundos na espera de uma descrição não é um valor baixo e foi reportado pelo usuário no momento dos testes. Apesar de não ter encontrado em nenhuma documentação do serviço, é possível que exista uma prioridade menor para requisições oriundas de contas do tipo gratuita (utilizada nos testes) comparado a contas pagas, acarretando em um tempo maior de espera para receber uma descrição.

A avaliação do modelo proposto apresentou algumas limitações: a descrição da imagem é apenas no idioma Inglês, a biblioteca (POD) (25) que implementa as funções de acesso a API não tem suporte em outro idioma; a dependência do aplicativo a um único serviço de descrição, caso esse serviço fique indisponível o aplicativo torna-se inutilizável. Uma configuração com a possibilidade do usuário escolher entre outros serviços de descrição eliminaria tal dependência. Para ouvir a descrição no idioma desejado pelo usuário, avaliou-se a possibilidade de enviar o texto descritivo em Inglês para um serviço remoto de tradução porém, aumentaria a média do tempo de espera da descrição e criaria mais uma dependência de serviço.

5. Conclusão

Atualmente áudioscrições estão se tornando bastante utilizadas em diversos ambientes de acesso à cultura, ainda que de uma forma limitadas ao espaço e conteúdo, auxiliam deficientes visuais no processo de inclusão social, tal inclusão traz autonomia e independência melhorando a qualidade de vida. Possibilitar que qualquer imagem, independente da pré análise de uma pessoa com visão normal, livre de limitações de conteúdo e espaço, como o presente trabalho propõem, pode comprovar a importância das pesquisas no campo da Tecnologia Assistiva bem como Redes Neurais e *Deep Learning*. Nas avaliações do protótipo concluiu-se que a média de trinta e três segundos para saber o que contém em uma fotografia é um valor aceitável levando em consideração o alto grau de assertividade do conteúdo da imagem. Como trabalhos futuros fica sugerido, permitir que o usuário opte por outros serviços remotos de descrição e a tradução para o idioma desejado.

Referências

- 1 PERON, M. *Tecnologia a favor das pessoas portadoras de necessidades especiais*. Disponível em: <https://www.tecmundo.com.br/software/2789-tecnologia-a-favor-das-pessoas-portadoras-de-necessidades-especiais.htm>.
- 2 STATISTA. *Number of available apps in the Apple App Store from July 2008 to January 2017*. Disponível em: <https://www.statista.com/statistics/263795/number-of-available-apps-in-the-apple-app-store/>.
- 3 STATISTA. *Number of available applications in the Google Play Store from December 2009 to March 2017*. Disponível em: <https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/>.
- 4 TÉCNICAS, C. de A. Ata vii reunião - cat corde / sedh / pr. *ATA Reunião*, Dezembro 2007. Disponível em: http://www.infoesp.net/CAT/_Reuniao_VII.pdf.
- 5 CÉREBRO, R. M. e. *Cinco Aplicativos Inovadores para Cegos*. Disponível em: http://www2.uol.com.br/vivermente/noticias/cinco_aplicativos_inovadores_para_cegos.html.
- 6 HYPENESS. *Be My Eyes: app permite "emprestar" seus olhos a um deficiente visual*. Disponível em: <http://www.hypeness.com.br/2015/01/be-my-eyes-app-permite-emprestar-seus-olhos-a-um-deficiente-visual/>.
- 7 ONLINE, E. *App promete ajudar deficientes visuais a "verem" o mundo*. 2017. Disponível em: <http://exame.abril.com.br/tecnologia/app-promete-ajudar-deficientes-visuais-a-enxergarem-o-mundo/>.

- 8 IDENTIFI. 2017. Disponível em: <http://getidentifi.com/>.
- 9 SOUSA, M. M. K. Uso de visão computacional em dispositivos móveis para o reconhecimento de faixa de pedestres. *Biblioteca Digital Brasileira de Computação*, 2012.
- 10 MATUSIAK P. SKULIMOWSKI, P. S. K. Object recognition in a mobile phone application for visually impaired users. *Human System Interaction (HSI)*, 2013.
- 11 MARÇAL, P. K. V. M. C. S. Audiodescrição no contexto da educação patrimonial. *Revista Memória em Rede*, v. 6, n. 10, 2014. Disponível em: <https://periodicos.ufpel.edu.br/ojs2/index.php/Memoria/article/view/9442/6176>.
- 12 BRAGA, A. P. d. L. F. d. C. e. T. B. L. Antônio de P. *Redes Neurais Artificiais - Teoria e Aplicações*. 2. ed 2007. ed. [S.l.]: LTC, 2007.
- 13 FERNEDA, E. Redes neurais e sua aplicação em sistemas de recuperação de informação. *Ci. Inf., Brasília*, v. 35, n. 1, p. 25–30, Janeiro 2006.
- 14 LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 5 2015. ISSN 0028-0836. Disponível em: <http://https://doi.org/10.1038/nature14539>.
- 15 MARINS, C. E. Desafios da informática forense no cenário de cloud computing. *Precedings of the Fourth Internacional Conference of Forensic Computer Science*, Julho 2009.
- 16 MICROSOFT. *Azure*. 2017. Disponível em: <https://azure.microsoft.com/en-us/services/cognitive-services/computer-vision/?v=17.23h>.
- 17 GOOGLE. *Cloud*. 2017. Disponível em: <https://cloud.google.com/vision/>.
- 18 IBM. *Watson*. 2017. Disponível em: <https://www.ibm.com/watson/developercloud/visual-recognition.html>.
- 19 CLOUDSIGHT. *CloudSight*. Disponível em: <https://cloudsight.ai>.
- 20 NOTE, W. W. G. *Web Services Architecture*. 2004. Disponível em: <https://www.w3.org/TR/ws-arch/#whatis>.
- 21 APPLE. *Swift 4 - The powerful programming language that is also easy to learn*. Disponível em: <https://developer.apple.com/swift/>.
- 22 LECHETA, R. R. *Desenvolvendo para iPhone e iPad - Aprenda a desenvolver aplicações utilizando o iOS SDK*. 5. ed. [S.l.]: Novatec, 2017.
- 23 APPLE. *VoiceOver Getting Started Guide*. 2017. Disponível em: <https://help.apple.com/voiceover/info/guide/10.12/>.
- 24 KEENAN, T. *Comparing Image Recognition APIs*. Disponível em: <https://www.upwork.com/hiring/data/comparing-image-recognition-apis/>.
- 25 FOLKENS, B. *ClouSightAPI*. Disponível em: <https://cocoapods.org/pods/CloudSight>.