Relatório Técnico - Android

Spotter - Image Classifier

Data: Aveiro, 5 de Abril de 2017

Autores: 71902: Pedro Ferreira de Matos
71770: Tiago Alexandre Lucas de Bastos

Resumo: O Spotter é uma aplicação móvel desenvolvida para reconhecer objetos em fotografias, essas fotografias são obtidas através de um Raspberry Pi 3 com o acessório da câmara, dispositivo esse que suporta o desenvolvimento na recente *framework* Android Things, para além do módulo que obtêm e classifica fotografias, com esta aplicação, é possível ver as fotografias obtidas e a sua respectiva classificação no telemóvel.

Índice

1 Introdução

Esta aplicação é feita a recurso a uma nova plataforma: o Android Things. O objetivo é de usar esta plataforma para tirar fotografias com uma câmara externa ao telemóvel e classificá-las através do Tensor Flow.

As imagens classificadas devem ser mostradas ao utilizador e haver um gráfico para poder agregar as classificações e permitir uma análise mais simples para o utilizador.

2 Âmbito e conceito da aplicação

Esta aplicação está orientada para todos os utilizadores que permitem obter o reconhecimento de objetos através de uma câmara estática.

Um dos cenários ideias seria uma obtenção de imagens de todas as pessoas que passaram num determinado sítio e a classificação poderia ser o reconhecimento de uma pessoa que já tivesse passado, de forma a contabilizar as pessoas que acediam a um determinado sítio.

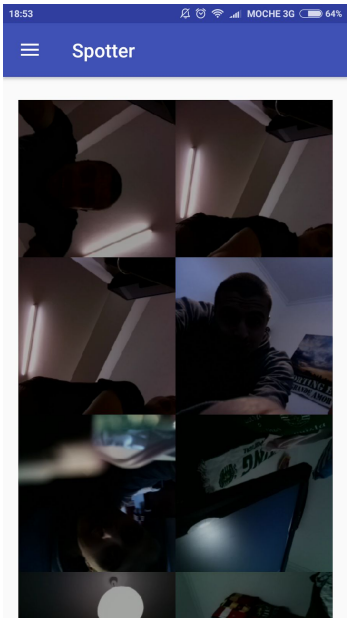
3 Detalhes técnicos

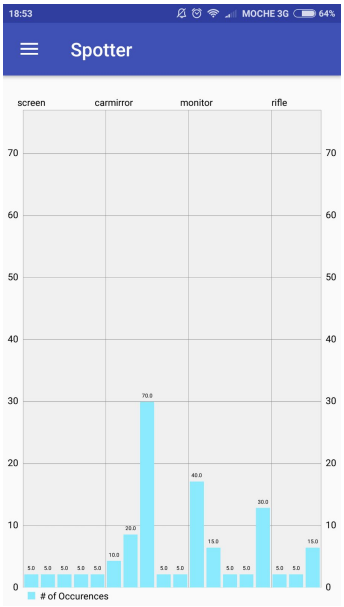
Design da Interface Visual

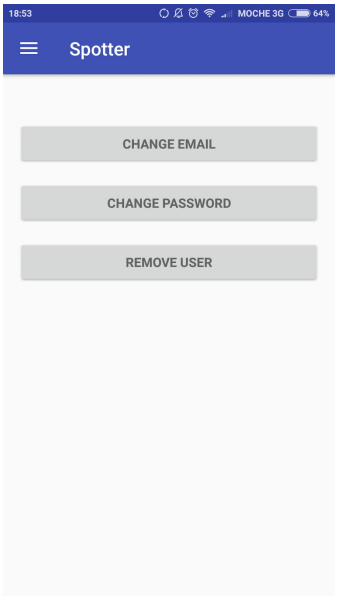
Fragmentos

Decidimos usar fragmentos em vez de múltiplas *Activities* em separado, devido ser um bom padrão de programação no Android, e também porque esses fragmentos estão todos interligados entre si, não havendo necessidade de se estar a "separar" a aplicação.

Esta decisão facilitou-nos o uso de um Navigation Drawer, visto que é comum a todos os fragmentos.



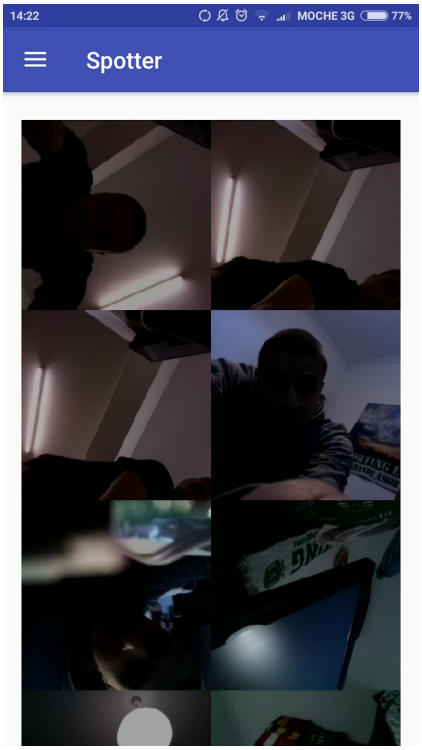




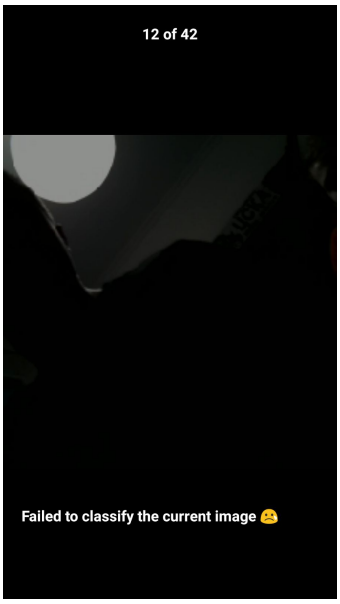
Galeria de Fotografias

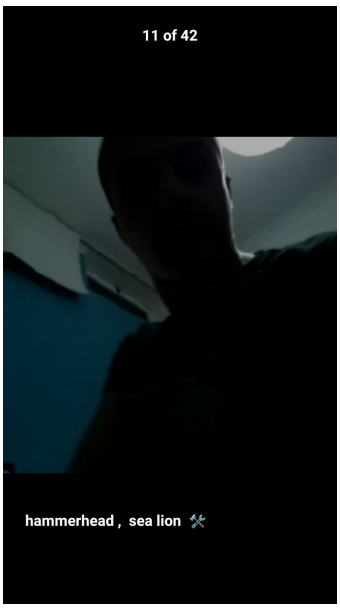
Como o foco principal da aplicação que corre no **telemóvel**, seria ver as fotos, o componente principal seria construir uma galeria que fosse fácil de utilizar e permitisse dar ao utilizador uma boa pré-visualização da imagem.

Desse modo, foi escolhido o componente ***RecyclerView***, que muito resumidamente é uma versão melhorada de uma ***listView***, com alguns ganhos de performance em scroll, visualização de Imagens, etc… .



Quando o utilizador clicar numa imagem presente no ***RecyclerView***, irá abrir uma "*View"* de FullScreen, onde poderá ver a imagem ampliada e também um texto contendo as classificações obtidas pelo Tensor flow.





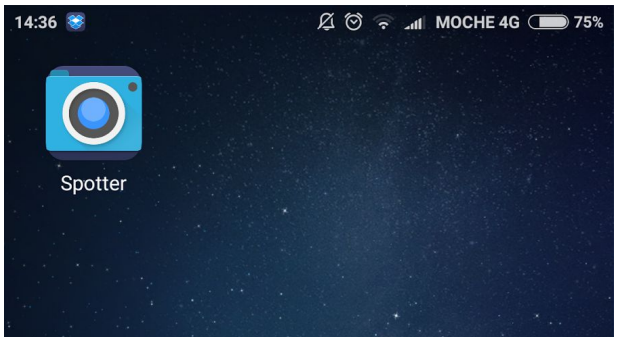
Cores

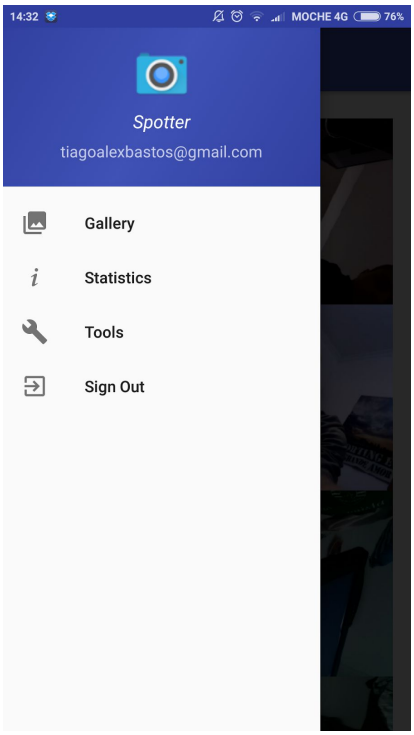
Esta aplicação foi desenvolvida maioritariamente em tons de azul, de forma a combinar com a cor do logo escolhido e também por ser uma cor agradável aos olhos não ferindo ou não sendo demasiado clara, desta forma conseguimos sempre manter um padrão de cor fixo por todos os componentes da aplicação mantendo todos os detalhes sempre bem visíveis.

Ícones

Os ícones desta aplicação foram escolhidos cuidadosamente de modo a se relacionarem a 100% com o texto que é suposto ilustrarem.

Ícones esses que foram obtidos sempre de forma a produzirem uma imagem 100% nítida, a melhor forma de o fazer, é obter um ícone com as dimensões necessárias com o tamanho da aplicação. Foram utilizados ficheiros *.xml* que definem ícones em vez dos tradicionais *.png* ou *.jpg*, porque produziam melhores resultados.



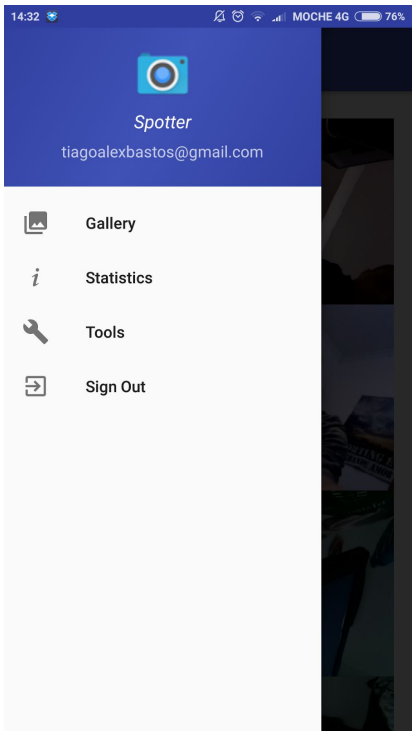


Outros Componentes

Outros componentes importantes de realçar são o:

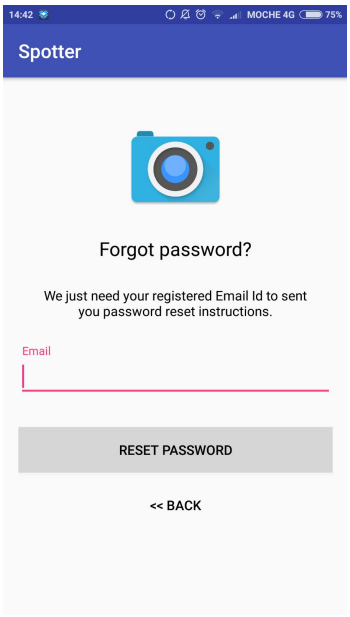
Navigation Drawer

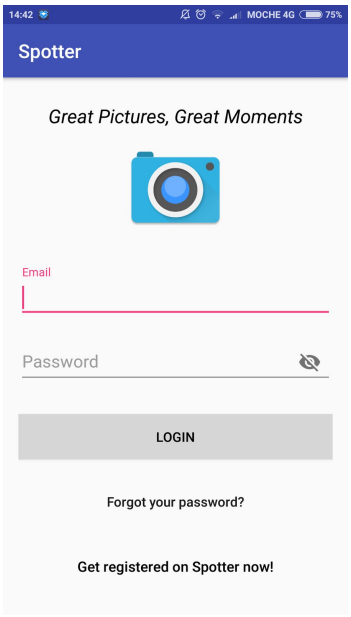
Um dos componentes principais de todas as vistas da aplicação, foi desenhado de modo a mostrar informação útil ao utilizador, como o seu nome de registo na aplicação, e é também composto pelo ícone e nome da aplicação, e botões que permitem ao utilizador saltar entre Fragmentos.

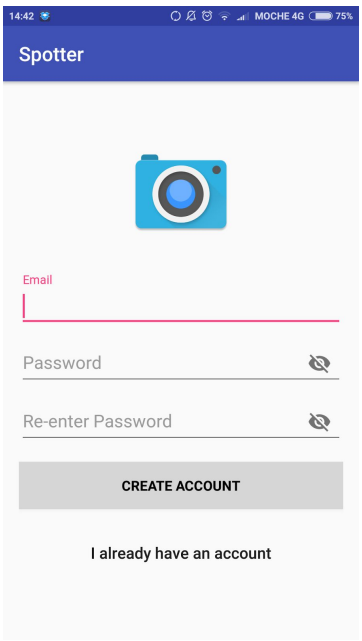


Ecrã de Login e Definições

Os ecrãs que compõem o Login e as definições são compostos por botões que ocupam a largura do telemóvel, que executam várias funções, funções essas que são facilmente identificáveis pelo texto que compõe cada botão





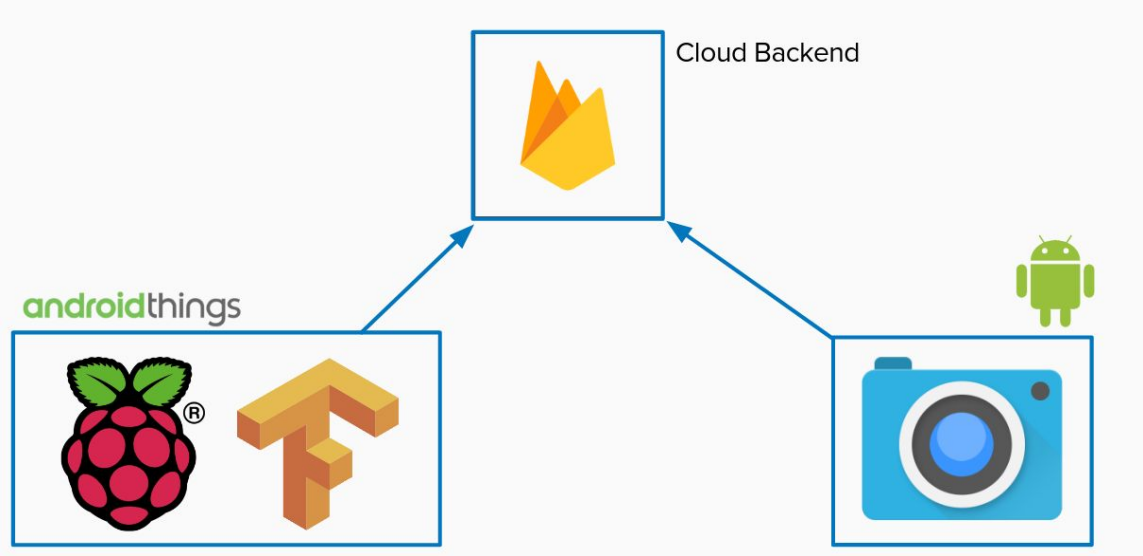


Ecrã de Estatísticas

Este ecrã é composto por um gráfico, novamente, composto por tons de azul de modo a enquadrar-se com a restante aplicação, é um gráfico de barras, onde é possível aumentar ou diminuir a escala consoante o toque no gráfico.

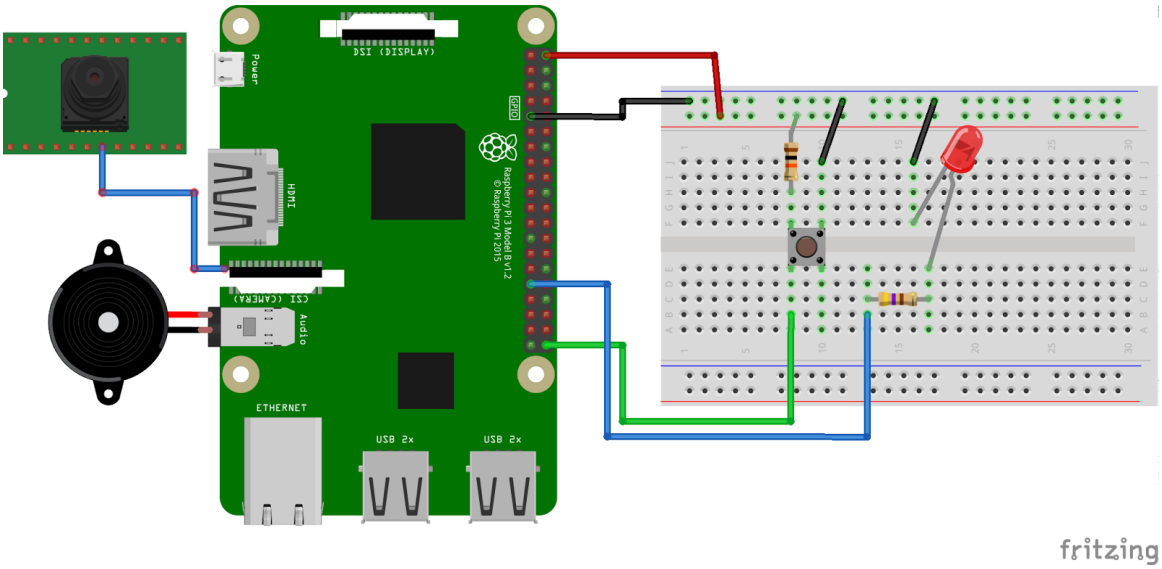
Design Técnico

Arquitetura



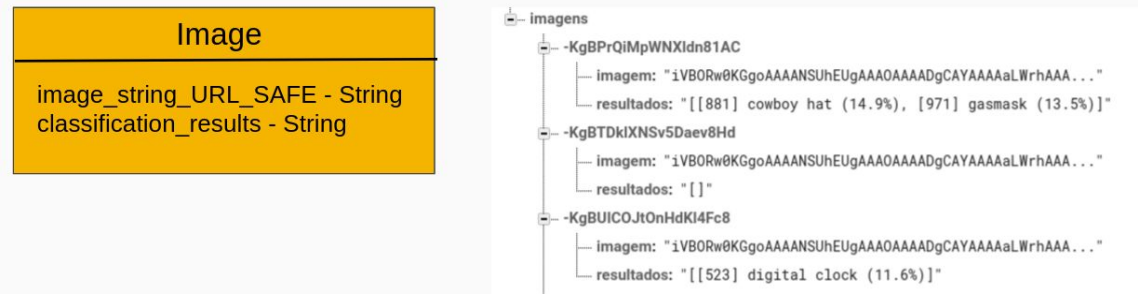
A imagem acima representa a Arquitetura global do nosso projeto, de um lado temos a parte de classificação e obtenção de imagem (Raspberry + TensorFlow) Android Things, o Back-End é controlado pela plataforma FireBase, e do lado contrário, temos a aplicação móvel Android, que será instalada nos telemóveis.

Circuito



O circuito utilizado no Raspberry Pi, foi o ilustrado na imagem acima, no nosso caso optámos por retirar o LED porque no âmbito do nosso projeto a interação com a placa não tinha um interesse muito elevado, tal como vamos ver na secção das soluções implementadas, a execução do projeto funciona sem recurso à placa.

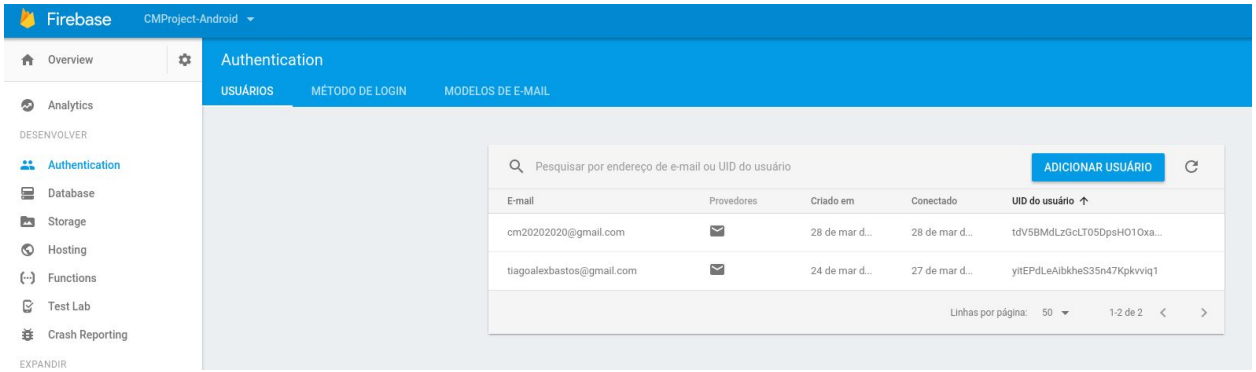
Modelo de Dados



A imagem acima corresponde ao modelo de dados da nossa aplicação, este modelo de dados foi implementado na *RealTime Database* presente no Firebase, deste modo conseguimos ter um armazenamento global à aplicação móvel e à aplicação Android Things.

Back-End

Toda a parte de Autenticação, gestão de utilizadores e como já foi referido armazenamento foi "colocado" no Firebase, achámos por bem achar esta ferramenta porque nos permitiu poupar um grande trabalho, caso não existisse acesso ao Firebase, iríamos ter de construir a nossa própria plataforma Web onde fosse possível realizar autenticação e armazenamento.



Persistência

Em relação a persistência, tal como já foi referido, aproveitamos a Base de dados do Firebase mas também o Internal Storage disponível no telemóvel. Os detalhes de implementação desta parte irão ser descritos na secção seguinte.

4 Soluções Implementadas

Android Things

É a tecnologia mais importante deste trabalho, o Android Things permite uma interação entre um ambiente Android e um chamado “mini-computador” tal como um Raspberry Pi, Intel Edison, etc…

Optámos por seguir 2 exemplos fornecidos pela página de Developers do Android Things para desenvolver a nossa aplicação final.

Essa aplicação, obtém fotos com o periférico Pi Camera, através do clique num botão presente na placa ou então de X em X segundos, no nosso caso 10.

A fotografia depois de obtida é enviada para a Base de Dados do Firebase associada à aplicação.

Enviado juntamente com a fotografia é também os resultados da classificação realizada pelo Classificador de Imagens, este classificador é uma rede neural, previamente treinada, que permite reconhecer objetos numa imagem, e foi desenvolvido com o Tensor flow.

O típico use-case do Android Things é o seguinte:

Android - Telemóvel

Com o telemóvel, conseguimos obter todas as imagens capturadas e classificadas no Raspberry Pi através do acesso à Base de Dados partilhada entre ambos, após se registar e fazer login na aplicação, o utilizador pode obter todas as fotografias, para isso apenas precisa de esperar que seja acabado o download das mesmas.

Existe uma feature bastante interessante em utilizar a base de dados do firebase, em que cada vez que é adicionada uma fotografia pelo Raspberry Pi á Base de Dados, é ativada uma callback no telemóvel que atualiza a galeria de novo, ou seja, a base de dados é novamente obtida de modo a mostrar as novas fotografias.

Se o utilizador não tiver internet é possível na mesma mostrar as fotografias que a aplicação já obteve, para isso, decidimos usar o Internal Storage como forma de persistência para que o utilizador consiga ter no seu telemóvel as fotografias que estavam presentes no firebase.

A estratégia implementada para fazer este componente foi a seguinte, de cada vez que a callback da base de dados é acionada, a pasta no internal Storage onde as fotografias se encontram é limpa e são colocadas na pasta as imagens obtidas, dessa forma eliminamos por completo a repetição de imagens, visto que a base de dados é sempre totalmente obtida. Desta forma, sempre que o utilizador não tiver acesso à internet, a galeria é preenchida por imagens que estão presentes no dispositivo, que correspondem às últimas obtidas aquando online. Quando volta a haver conexão, a galeria de imagens é novamente atualizada.

Um dos outros componentes com interesse ao utilizador final é o fragmento onde é mostrado um gráfico de barras com o número de ocorrências de cada classificação, para construir este componente, aproveitamos mais uma vez a callback da base de dados, de modo a que cada vez que haja uma fotografia nova, o gráfico seja novamente desenhado.

Quando o utilizador está offline, pode carregar o último gráfico construído do momento em que esteve online, para isto, é novamente utilizado o Internal Storage, onde é gravado o objeto que representa um Mapa<Resultado, Nº Ocorrências>, que irá ser utilizado para desenhar o gráfico novamente. Mal o telemóvel obtenha conexão de novo, a callBack da base de dados é novamente acionada, populando o gráfico com a mais recente informação.

Em termos de testes, não foram utilizados testes unitários, mas a aplicação foi testada com o Monkey (500 toques) e foi feita a análise do código pelo Link disponibilizado pelo Android Studio.

6 Conclusão

O objetivo do projeto seria o reconhecimento de uma pessoa de forma a que estes dados pudessem ser usados para estatísticas sobre a contabilização de pessoas. No entanto, ao realizarmos o projeto apercebemo-nos de que a aplicação que deveria fazer a classificação de imagens não consegue reconhecer seres humanos/pessoas, apenas objetos. Desta forma, tivemos que reconsiderar a ideia inicial e a aplicação será um agregador de imagens que obtêm uma classificação, e as quais são apresentadas ao utilizador. No entanto, esta classificação apenas é utilizada para objetos e não pessoas.

No decorrer do desenvolvimento do projeto, após várias pesquisas, descobrimos que o Tensor flow disponibiliza de facto um reconhecedor de pessoas, mas após várias tentativas de adaptar o reconhecedor de pessoas ao Android Things acabámos por decidir manter o classificador de objetos que é disponibilizado nos exemplos do Android Things.

Apesar dos problemas encontrados com a classificação do Tensor Flow, não houve problemas significativos com o uso do Android Things, o que nos permitiu conhecer mais uma plataforma e expandir os nossos conhecimentos. Desta forma, ficamos apenas desiludidos devido à incapacidade do Tensor Flow em classificar pessoas e com o facto de a resolução da câmara do Raspberry Pi ser muito pequena, o que leva a grandes dificuldades na classificação dos objetos.

Com estes problemas em mente apenas recomendamos, se possível, a obtenção de uma câmara de melhor qualidade para a facilitação da classificação de imagens por parte do software de classificação, neste caso o Tensor Flow.

7 Referências e Recursos

Raspberry Pi, Get Started - Android Things
Android Things - Samples
Recyclerview
Firebase Android Setup
Internal Storage - Android
Emoji Library
TensorFlow - Image Classification

Recursos:

- Repositório: https://github.com/tiagoalexbastos/ProjetoAndroidCM
- .APK: https://github.com/tiagoalexbastos/ProjetoAndroidCM/blob/master/app-debug.apk