



## Auxílio a Deficientes Visuais utilizando Redes Neurais Convolucionais

Gabriel Minguini SANGA<sup>1</sup>; João Mauricio Gallego POLO<sup>1</sup>; Jefferson Antônio Ribeiro Passerini<sup>2</sup>

### RESUMO:

Atualmente no mundo existem 2,2 bilhões de pessoas que possuem alguma deficiência visual, e uma das dificuldades é auxiliar essa população no seu ato diário de se locomover, assim, o desenvolvimento de ferramentas que auxiliem essa população se faz necessário. Este projeto propõe a criação de um framework envolvendo um aplicativo móvel para captura de imagens em tempo real e feedback ao usuário através de “beeps” em relação ao caminho estar livre ou com obstáculos. As imagens capturadas são enviadas a um servidor onde ocorre a extração de características utilizando redes neurais convolucionais (VGG16 + VGG19) previamente treinadas pela base de imagem ImageNet, e a classificação utilizando o modelo de aprendizado supervisionado (SVM). Como resultado dos testes realizados obteve-se uma acurácia de 75,3% das imagens capturadas pelo aplicativo.

### PALAVRAS-CHAVE:

Detecção de obstáculos. Redes neurais convolucionais. Transferência de aprendizado. Aprendizado supervisionado. Deficientes Visuais.

### ABSTRACT:

*Currently in the world there are 2.2 billion people who have some kind of visual impairment, and one of the difficulties is to help this population in their daily act of moving around, thus, the development of tools that help this population is necessary. This project proposes the creation of a framework involving a mobile application to capture images in real time and feedback to the user through “beeps” in relation to the path being free or with obstacles. The captured images are sent to a server where feature extraction takes place using convolutional neural networks (VGG16 + VGG19) previously trained by the ImageNet image base, and classification using the supervised learning model (SVM). As a result of the tests carried out, an accuracy of 75.3% of the images captured by the application was obtained.*

### KEYWORDS:

*Obstacle detection. Convolutional neural networks. Learning transfer. Supervised learning. Visually Impaired.*

<sup>1</sup>Acadêmicos do 4º Ano do Curso de Graduação em Sistemas de Informação das Faculdades Integradas de Fernandópolis – FIFE-FEF, Fernandópolis-SP.

<sup>2</sup> Professor, Mestre em Ciências da Computação, Professor do Curso de Graduação em Sistemas de Informação das Faculdades Integradas de Fernandópolis – FIFE-FEF, Fernandópolis-SP.

## 1. Introdução

Atualmente em nossa sociedade, pessoas com deficiência visual ou totalmente cegas se tornarão um aspecto muito importante. Sendo um problema para as instituições governamentais cada vez mais eminente, visto que, os mesmos carecem de planejamento e artifícios para contornar e minimizar os danos provenientes.

De acordo com a ONU (2019), existem cerca de 2,2 bilhões de pessoas que possuem algum tipo de deficiência visual, onde a assistência às essas pessoas é escassa. Fato este que é agravado pela falta de atendimento igualitário e baixa qualidade de serviços prestados.

Segundo o IBGE (2010), no censo demográfico realizado em 2010, o Brasil registrou mais de 35 milhões de pessoas que declaravam possuir algum acometimento que prejudicava em algum grau sua visão. Desses, 506.377 eram totalmente cegos e outros 6.056.533 possuíam grande dificuldades em enxergar.

Dessa população de totalmente cegos, 27,26% se encontravam na faixa acima dos 65 anos, 59,63% entre 15 e 64 anos e 13,11% em menores de 14 anos. Se observar a população que se declarava com grande dificuldade de visão, 29,44% da população acima de 65 anos e da população entre 15 e 64 anos, encontra-se um índice de 65,65% e, na população menor de 14 anos, têm-se 4,91% IBGE (2010).

Quando se fala em deficiência visual no Brasil e no Mundo, aborda de uma situação que afeta a qualidade de vida das pessoas que sofrem e dos familiares a sua volta, causando muitas vezes, desequilíbrio social se não tratada adequadamente. Os problemas citados levaram vários pesquisadores a explorar novos caminhos em várias disciplinas, como tecnologias assistivas, psicologia cognitiva, visão computacional, processamento sensorial, reabilitação, acessibilidade incluindo interação humano-computador.

As tecnologias assistivas facilitam o acesso dos deficientes visuais a informações, promovem a segurança, apoiam sua mobilidade e geram considerável melhora na qualidade de vida, com grande impacto direto na inclusão social (MANDUCHI; COUGHLAN, 2012).

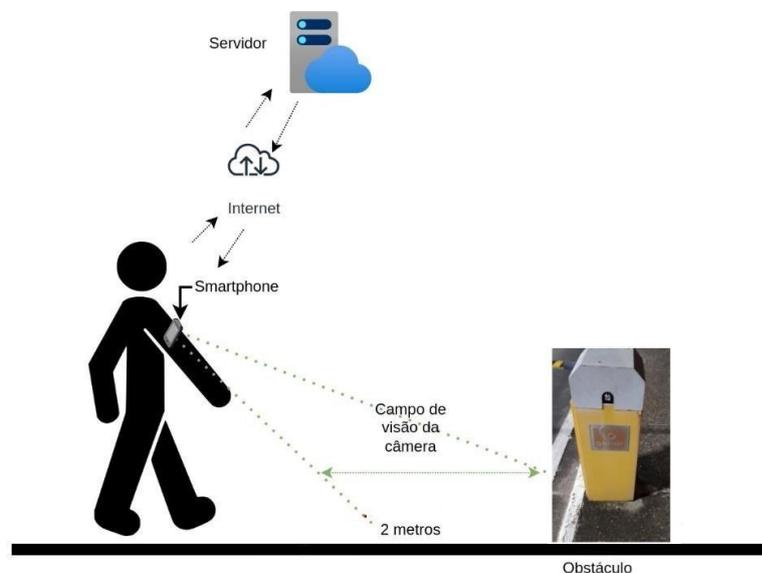
Dessa forma, considerando o cenário atual, vislumbra-se um sistema que permita aos deficientes visuais aproveitar os recursos disponíveis em seu smartphone sem recursos extras ou requisitos de conexão.

Podendo funcionar com seu aparelho tirando fotos do caminho que a pessoa está percorrendo e fornecendo um feedback por áudio ou vibração a fim de demonstrar que há obstáculos a frente, antes mesmo que esses cheguem ao alcance do *smartphone*.

## 2. Metodologia

O framework proposto por este trabalho é composto recursos para auxiliar a navegação de uma pessoa com deficiência visual. Na Figura 1 observa-se a utilização por um usuário do aplicativo, onde o sistema emitirá alertas de áudio no *smartphone* com aviso antecipado a obstáculos.

**Figura 1** – Esquema de trabalho do framework.



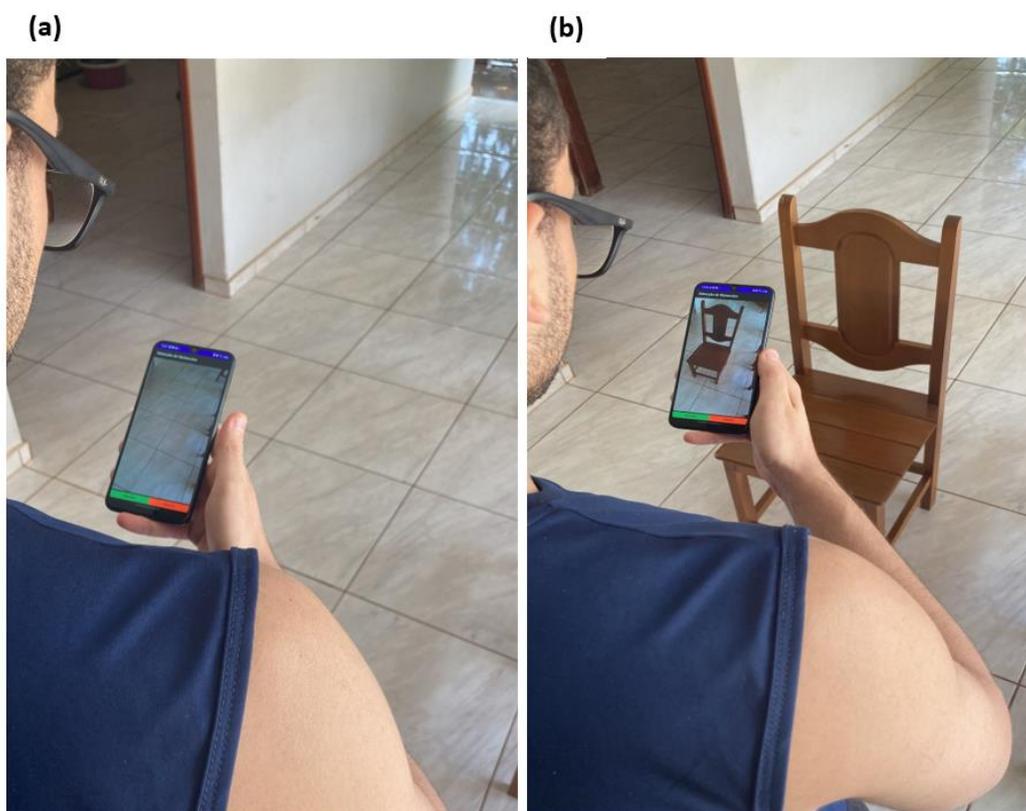
Fonte: Os Autores (2022).

O sistema emitirá alertas sonoros no *smartphone* quando detectar obstáculos por meio da coleta de *frames* de imagem a partir da câmera do aparelho. As informações coletadas no *smartphone* são processadas externamente em uma API *Flask* (em um servidor web) onde tem-se uma rede neural convolucional treinada a partir da ImageNet utilizando *transfer learning* e um classificador SVM previamente treinado com imagens exemplos de obstáculos, o modelo realizará o processamento da imagem retornando a informação de caminho livre ou com obstáculos ao *smartphone*, que retornará o feedback ao usuário através de “*beeps*”, onde um sinal sonoro representa caminho livre e dois sinais sonoros seguidos significa obstáculos a

frente. A comunicação entre o smartphone e o servidor pode ocorrer por qualquer rede TCP/IP inclusive por meio de uma conexão via Internet.

Na Figura 2 visualiza-se como será o funcionamento real do aplicativo, onde em (a) observa-se a detecção de um caminho sem obstáculos e em (b) temos a observação pelo aplicativo de um obstáculo.

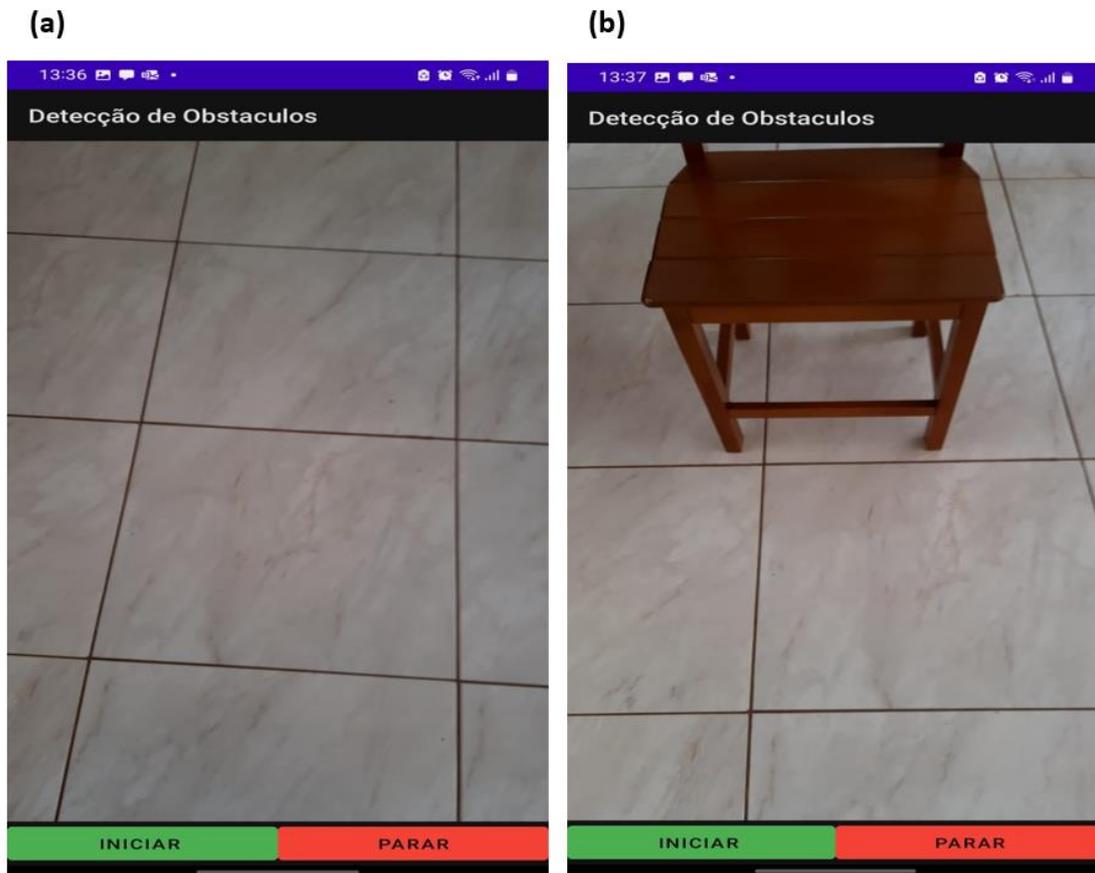
**Figura 2** – Demonstração da utilização do protótipo desenvolvido: (a) Sistema detectando caminho sem obstáculos e (b) Sistema com obstáculos no caminho do usuário.



Fonte: Os Autores (2022).

Na Figura 3 verifica-se mais detalhadamente a interface do aplicativo, na Figura 3(a) o aplicativo tem caminho livre a frente e na Figura 3(b) observa-se um obstáculo onde o aplicativo deverá alertar seu usuário com a emissão de dois sinais sonoros seguidos.

**Figura 3** – Demonstração real do funcionamento do aplicativo em um percurso com obstáculos a frente.



Fonte: Os Autores (2022).

A coleta de imagens pelo aplicativo utilizará a câmera existente no smartphone, o que deverá ser realizado com o smartphone na posição vertical, aumentando a amplitude da imagem à frente. A imagem coletada tem por objetivo mapear o caminho à frente do usuário para proporcionar um alerta antecipado.

Para que o alerta ocorra, é necessário que o serviço de dados esteja funcionando, onde ao enviar as imagens para o servidor Flask, onde ocorrerá a classificação da imagem como caminho limpo ou caminho com obstáculos, e, assim devolva uma resposta para o aplicativo Android, onde o mesmo irá interpretar e emitir sinais diferentes para os dois tipos de caminho, citados anteriormente.

Detalhando-se o processo descrito anteriormente, o modelo de classificação implementado está composto por duas etapas, extração de características e a classificação dessas características utilizando o modelo de aprendizado supervisionado SVM.

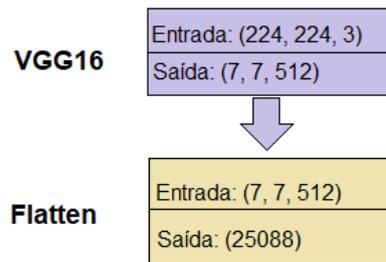
A primeira etapa constitui a extração de características através da utilização de redes neurais convolucionais (CNN), sendo a ascensão dos métodos de aprendizagem profunda a responsável por avanços no processo de reconhecimento de objetos e classificação de imagens (LECUN; BENGIO; HINTON, 2015; SCHMIDHUBER, 2015; HOWARD *et al.*, 2017; CAI *et al.*, 2016; KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

O processo de aprendizado de uma CNN quando se fala de processamento de imagens requer uma grande quantidade de amostras rotuladas para que ela possa estimar milhões de parâmetros. Sendo essa uma tarefa custosa e demorada, assim barrando a aplicação de CNNs em problemas quando se tem dados de treinamento limitados.

Deste modo para o processo de extração de características das imagens, as CNNs utilizadas não foram treinadas de forma específica para o problema proposto pelo trabalho, mas foi utilizado o processo *de transfer learning* com parâmetros pré-treinados a partir do conjunto de dados ImageNet. OQUAB *et al.* (2014) demonstraram que representações de imagens aprendidas com CNNs em conjuntos de dados rotulados em grande escala podem ser eficientemente transferidos para outras tarefas de reconhecimento visual com um limitado conjunto de imagens para treinamento.

Como proposta de se utilizarem as CNNs como ferramenta de extração de características das imagens, foram utilizadas as informações da última camada convolucional planificada (flatten layer) sem o emprego de uma camada de *pooling* e remove-se também a camada *fully-connected* responsável para a classificação na CNN, já que esse não é o seu objetivo. Tomando-se por exemplo em uma CNN VGG16 (SIMONYAN; ZISSERMAN, 2015) em sua última camada convolucional, tem-se uma matriz 3D ( $7 \times 7 \times 512$ ), que resulta em um vetor de características com 25088 informações (Figura 4).

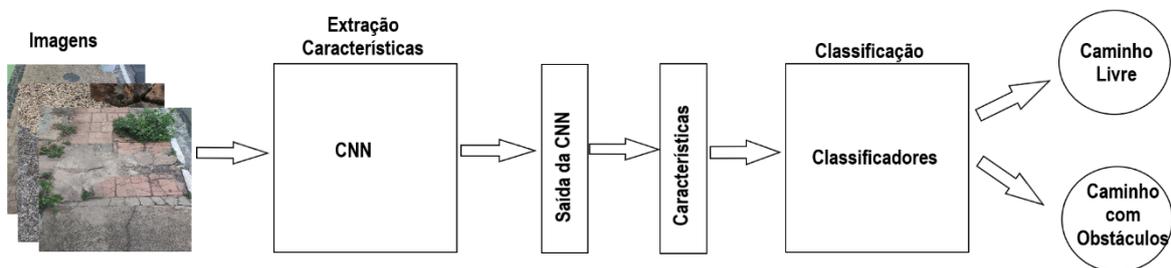
**Figura 4** – Exemplo de processo de extração de características (VGG16)



Fonte: Os Autores (2022).

No modelo proposto utilizou-se as redes neurais convolucionais VGG16 e VGG19, como extratores de características e como descrita anteriormente uma camada de classificação (Fully Connected), utilizou-se as CNNs a partir da implementação disponível no TensorFlow (ABADI *et al.*, 2015). Para ambas CNNs utilizou-se o conjunto de imagens disponibilizado pelo ImageNet (RUSSAKOVSKY *et al.*, 2015) para compor as camadas de convolução com pesos pré-treinados através de *transfer learning*. Sendo que, a última camada convolucional de ambas as arquiteturas, gerem matrizes de tamanho (7 x 7 x 512); deste modo, há um total de 25.088 características extraídas cada, pode-se ter uma visão inicial do método proposto na Figura 5.

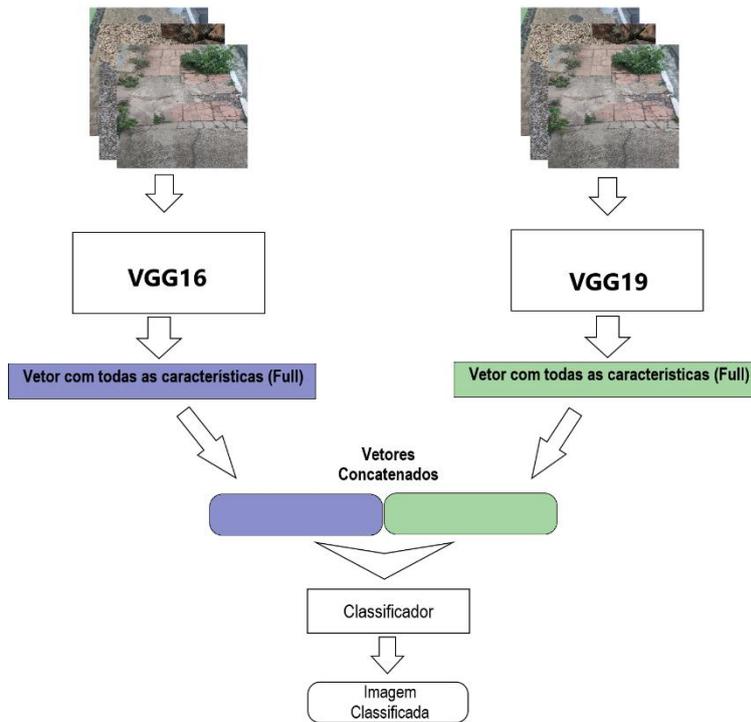
**Figura 5** – Modelo geral proposto



Fonte: Os Autores (2022).

Nesta abordagem, utilizamos combinações de CNNs para a extração de características, os vetores de saída de cada uma das redes convolucionais são concatenados e aplicados ao classificador, concatenando-se os vetores de saída das CNNs VGG16 e VGG19 teremos uma matriz de 50.176 características onde essas características são submetidas a um classificador treinado com dados alinhados ao propósito do trabalho (Figura 6).

**Figura 6** – Exemplo demonstrando a proposta da abordagem CNNs e processo de concatenação de vetores.



Fonte: Os Autores (2022).

A segunda etapa deve-se realizar o treinamento do classificador de Máquina de Vetor de Suporte (SVM) utilizando-se uma base de imagem específica com obstáculos com as mesmas características das imagens que se pretende coletar e classificar na aplicação em seu real funcionamento. O kernel utilizando na SVM foi o linear (LORENA; CARVALHO, 2003, PINEDA *et al.*, 2015) e o classificador foi executado a partir das configurações recomendadas no SciKit-Learn (PEDREGOSA *et al.*, 2011).

A base de imagens utilizada foi proposta por BREVE e FISCHER (2020) é composta por 342 imagens divididas em duas classes: 175 demonstram caminho livre e 167 caminho com obstáculos. Essas imagens foram obtidas com a câmera de um smartphone e redimensionadas para 750x1000 pixels. O smartphone foi colocado na altura do peito do usuário e inclinado aproximadamente de 30° a 60° em relação ao solo, para que pudesse capturar alguns metros à frente do caminho a ser percorrido.

Embora não seja grande, o conjunto de dados cobre áreas internas e externas, com diferentes tipos de piso (seco ou molhado), diferentes situações de iluminação e diferentes tipos

de obstáculos como cones, buracos, animais entre outros. Na Figura 7 pode-se visualizar exemplos de imagens existentes.

**Figura 7** – Imagens extraídas do conjunto de dados propostos para treinamento do classificador SVM



(a) Caminho Limpo

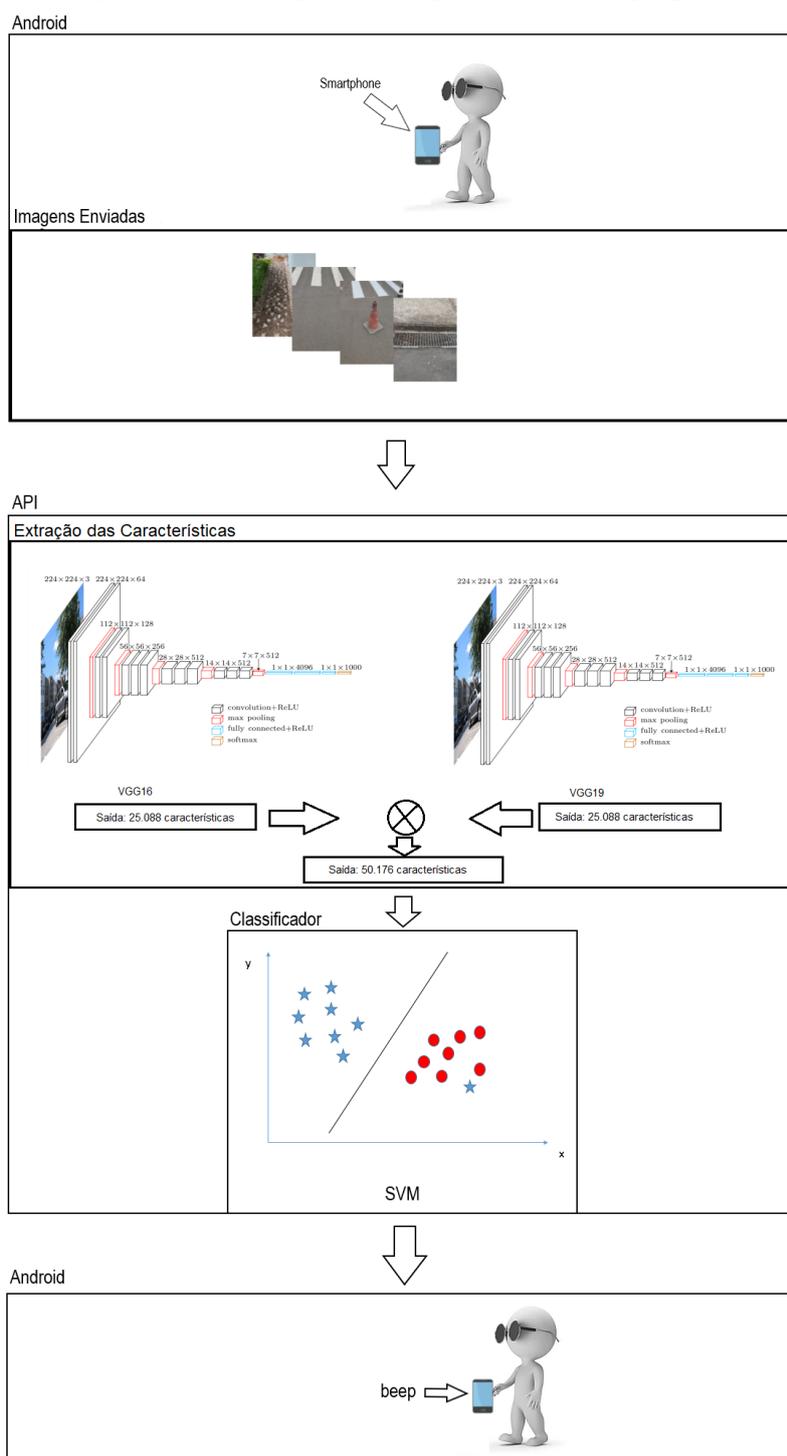


(b) Caminho com obstáculos

Fonte: Adaptado de BREVE e FISCHER (2022).

O classificador será executado em uma API Flask onde irá receber imagens em tempo real por meio de um aplicativo Android com acesso a câmera, ao receber essas imagens, a API com nosso classificador SVM já previamente treinado irá identificar se a imagem possui um obstáculo ou não, devolvendo esse retorno para o aplicativo Android, que deverá emitir um alerta auditivo para que o usuário desvie do mesmo. Podemos verificar o método proposto de forma resumida na figura 8.

**Figura 8** – Ilustração das etapas do método proposto.



Fonte: Os Autores (2022).

### 3. Resultados e Discussão

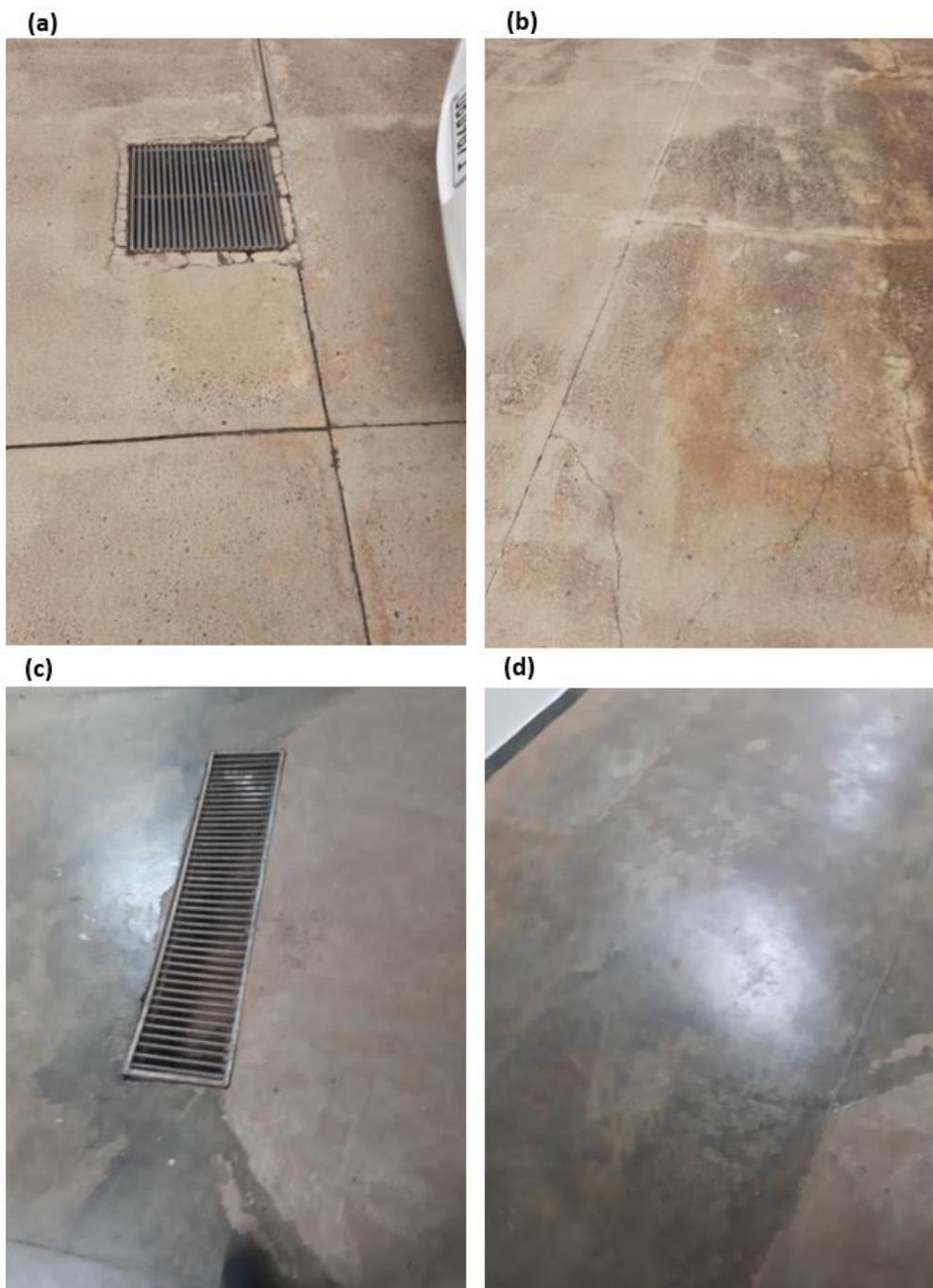
Para a execução dos testes criou-se um ambiente simulado de comunicação de rede entre o dispositivo móvel (*smartphone*) e o servidor *flask* (*notebook*) utilizando-se um roteador

utilizando a rede sem fio, deste modo a comunicação de dados entre os dispositivos envolvidos nos testes ocorreu em um ambiente de rede local o que certamente privilegia a eficiência de comunicação entre os dispositivos envolvidos no teste.

Realizou-se a avaliação de desempenho da abordagem proposta por meio das imagens que foram capturadas durante o percorrer do usuário, totalizando 194 imagens. Neste conjunto de imagens coletadas encontra-se imagens de diferentes locais com diferentes graus de luminosidade e alguns obstáculos, sendo que, preferencialmente partiu-se de uma maior atenção para obstáculos não presentes na base de treino, porém com características semelhantes, como profundidade e cores. Os testes foram realizados pelos próprios autores deste trabalho e deve-se enfatizar que a abordagem proposta não substitui a bengala e sim serve como complemento ao deficiente visual junto aos recursos de acessibilidade já disponíveis a ele.

Pode-se verificar alguns exemplos de imagens coletadas durante o processo de testes na Figura 9, onde tem-se imagens com obstáculos e sem obstáculos com iluminação natural e artificial.

**Figura 9** – Imagens coletadas: a) Imagem com obstáculo com luz natural b) Imagem sem obstáculos com luz natural c) Imagem com obstáculo com luz artificial d) Imagem sem obstáculos com luz artificial.



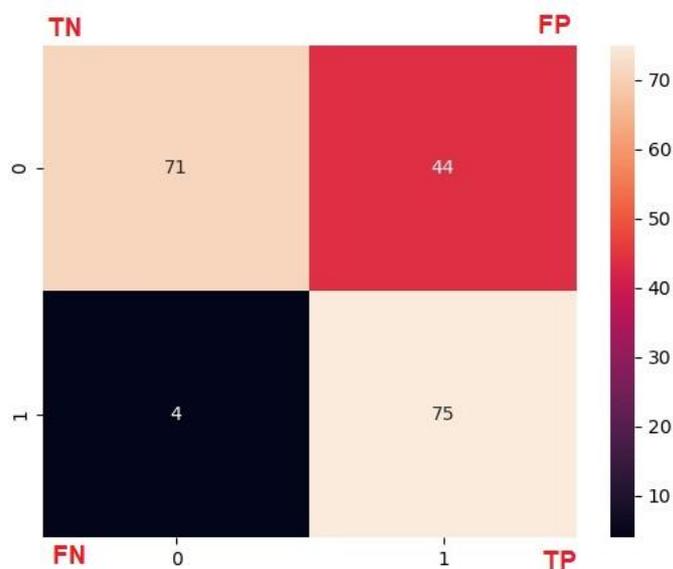
Fonte: Os Autores (2022).

Para aferição dos resultados as imagens recebidas pelo servidor a partir do envio do *smartphone* foram armazenadas e codificadas de forma a manter a correlação do resultado aferido pela predição do modelo de inteligência artificial aplicado, esse resultado também foi

armazenado. Ao final do processo de caminhar utilizando o dispositivo e de coletar as informações analisou-se e classificou-se as imagens como caminho com obstáculo ou sem obstáculo, esse processo é manual e foi realizado imagem a imagem. Uma vez possuindo-se as imagens devidamente classificadas efetuou-se a comparação entre o resultado da classificação manual com o resultado que foi obtido pela predição do algoritmo.

A partir da análise dos dados preparou-se uma matriz de confusão para demonstrar o resultado, onde os casos verdadeiros são as imagens que possuem obstáculos e falsos determinam são os de caminho livre, como pode-se observar na Figura 10.

**Figura 10** – Matriz de confusão.



Fonte: Os Autores (2022).

Os resultados obtidos foram influenciados por fatores externos e internos, sendo os externos a qualidade da imagem obtida pelos dispositivos (Distância, Luminosidade, Tonalidade de cor entre o chão e o obstáculo), velocidade da caminhada do usuário e a estabilidade do dispositivo na mão do usuário, como por exemplo na imagem abaixo onde pode-se visualizar a comparação de uma imagem com boa qualidade e outra com uma baixa qualidade. Na Figura 11 pode-se observar situações exemplos de imagens coletadas.

**Figura 11** – Demonstração de imagem capturada pelo aplicativo: (a) imagem com uma boa qualidade e visibilidade e (b) imagem com uma baixa qualidade e visibilidade.



Fonte: Os Autores (2022).

Deste modo temos uma acurácia calculada de 75,3% de acertos, e uma especificidade que apura o percentual de acertos de negativos entre os casos negativos de 61,7% e uma sensibilidade que mede a taxa de acertos de casos positivos entre todas as amostras positivas de 94,9%, ou seja, o modelo proposto de acordo com os testes realizados nos demonstra uma melhor detecção em situações que a imagem capturada possui algum obstáculo. Verificandose a eficiência que é a média entre as taxas de sensibilidade e especificidade temos uma taxa de 78,3%.

Quando se observa a precisão, temos que o modelo demonstra ser menos preciso com uma taxa de 63,02%. Pode-se verificar também as taxas de falso positivo: 38,3% e de falso negativo: 5,06%, ou seja, novamente temos um número que reforça a melhor assertividade do modelo para casos em que as imagens coletadas possuem obstáculos.

Apurando-se o coeficiente de correlação de Matthews onde -1 demonstra uma não correlação e +1 uma correlação perfeita, obtêm-se um índice de correlação de 0,61 para a classificação efetuada no modelo computacional utilizado, esse índice de correção determina

uma correlação substancial (0,50 e 0,69) entre o resultado da classificação realizada e os elementos analisados imagens com caminho limpo e com obstáculos.

Na Figura 12(a) e Figura 12(b) apresenta-se amostras de imagens com caminhos com obstáculos que foram corretamente classificadas, nas Figura 12(c) e Figura 12(d) temos imagens com caminhos limpos corretamente classificadas.

**Figura 12** – Amostras com classificação correta: (a) e (b) imagem com caminho com obstáculos e (c) e (d) imagens com caminho limpo.



Fonte: Os Autores (2022).

Na Figura 13(a) e Figura 13(b) apresenta-se amostras de imagens com caminhos com obstáculos que foram erroneamente classificadas, nas Figura 13(c) e Figura 13(d) temos imagens com caminhos limpos erroneamente classificadas.

**Figura 13** – Amostras com classificação correta: (a) e (b) imagem com caminho com obstáculos e (c) e (d) imagens com caminho limpo.



Fonte: Os Autores (2022).

Analisando-se aspectos referentes ao processo de treinamento do classificador a ampliação da base de imagens proposta por BREVE e FISCHER (2020) pode ser um fator relevante para a melhora no desempenho do classificador. Outro aspecto relevante foi o processo de transferência de aprendizado aplicado a rede neural convolucional (CNN) responsável pela extração das características, foi utilizado a base ImageNet que apesar de possuir muitas imagens não tem como objetivo o uso de identificação de objetos/obstáculos, a busca por uma base mais apropriada para o fim proposto poderia contribuir também com a melhoria de desempenho.

A aplicação de outras arquiteturas de CNNs para extração de características também pode contribuir com a melhoria da identificação, assim como a utilização de recursos externos adicionais como sensores.

Em relação ao desempenho da aplicação mobile depende diretamente da arquitetura (processador e memória) do *smartphone* utilizado, observou-se pequenos travamentos na visualização do frame de imagem capturado durante a filmagem da cena pelo dispositivo, aparelhos com hardware mais potente contribuem para a fluidez da execução.

A abordagem apresentada também precisa ser testada utilizando a internet, através da rede 4G, para realizar a comunicação com o servidor que deve estar hospedado em algum serviço na web disponível ao uso geral. A rede de celular pode ser um fator complicador em relação ao envio do frame capturado e o retorno da classificação ao *smartphone* causando pequenos “*delays*” para o usuário. A chegada da rede 5G deve minimizar grandemente esse possível problema devido ao seu alto poder de transferência de dados.

Outro aspecto a ser observado é em relação ao acesso do deficiente visual aos planos de dados das operadoras de celular, que apesar de estarem a cada dia mais populares, ainda são custosos para muitas pessoas em nosso país. Assim outra vertente de pesquisa é a implementação de classificadores e extratores de características que possam ser executados de forma nativa nos recursos ainda limitados de um *smartphone*.

Deve-se também observar a necessidade da continuidade da pesquisa devido a necessidade de realizar-se mais testes em diferentes ambientes e condições e principalmente com o uso de bengala e em condições mais reais (apesar de simuladas) de um usuário que enxerga com os olhos vendados conseguir navegar em um percurso pré-definido, assim como trazer para pesquisa associações de apoio a deficientes visuais que possam contribuir com o processo de testagem do aplicativo.

#### 4. Considerações Finais

Neste trabalho apresentou-se um framework desenvolvido para o programa de iniciação científica da FIFE/FEF, onde obteve-se a predição de resposta de caminho livre ou com obstáculos em auxílio a navegação de deficientes visuais. Esse processo se deu a partir da captura de frames de imagem através de um smartphone, o envio a um servidor onde um algoritmo de inteligência artificial realizou a classificação e o seu retorno ao dispositivo móvel que notificou o usuário através da utilização de *beeps*.

Considerando-se os métodos e abordagens utilizadas, percebeu-se que apesar da taxa de acurácia de 75,3% pode-se implementar outras abordagens de algoritmos, sensores e tecnologias para a busca de melhoria na predição desejada.

A aplicação de arquiteturas de CNN mais novas ao modelo deve contribuir com ganhos de desempenho na classificação das imagens, assim como a integração de sensores a uma bengala pode auxiliar ainda mais neste processo. Visualizou-se a necessidade de aprofundamento dos testes, simulando situações de utilização cada vez mais próximas das reais.

Percebeu-se a necessidade de aumentar a base utilizada para treinar o classificador e diversificar as imagens de obstáculos, aumentando a distância deles em relação ao observador. Nota-se também que, em alguns smartphones há um *beep* de atraso no aplicativo, onde o fato se dá em relação a como cada sistema trata a captura de imagens em tempo real.

Finalmente pode-se afirmar que foi possível implementar e testar a abordagem proposta com um grau de sucesso e que se deve buscar a continuidade da pesquisa buscando melhorar e aproximar o framework do seu uso real, pois a sua utilização poderia auxiliar milhares de pessoas no Brasil e no mundo, na simples atividade de se locomover.

#### 5. Referências

1. BREVE, F.; FISCHER, C. N. Visually impaired aid using convolutional neural networks, transfer learning, and particle competition and cooperation. International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, March 2020.
2. CAI, Z. et al. A unified multi-scale deep convolutional neural network for fast object detection. European Conference on Computer Vision ECCV, Part IV, Springer, p. 354-370, 25 Jul, 2016.
3. HOWARD, A. G. et al. Mobilinets: efficient convolutional neural networks for mobile vision applications, arXiv:1704.04861v1 [cs.CV], 17 Apr 2017.

4. IBGE, BRASIL. Censo Demográfico 2010. Instituto Brasileiro de Geografia e Estatística. Disponível em: <[https://biblioteca.ibge.gov.br/visualizacao/periodicos/94/cd\\_2010\\_religiao\\_deficiencia.pdf/](https://biblioteca.ibge.gov.br/visualizacao/periodicos/94/cd_2010_religiao_deficiencia.pdf/)>, Acesso: 05/08/2022. Brasília-DF, 2010.
5. KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: 25th International Conference on Neural Information Processing Systems. Advances in Neural Information Processing Systems 25 (NIPS, 2012), December 2012. Proceedings... Curran Associates, p. 1097-1105, 3 December 2012.
6. LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. Nature, v. 521, n. 7553, p. 436-444, 28 May 2015.
7. LORENA, A. C; CARVALHO, A. C. P. L. F. Introdução máquinas de vetores de suporte. Relatório Técnico do Instituto de Ciências Matemáticas e de Computação [192]. São Carlos, SP: ICMS-USP, 2003. 66 p.
8. MANDUCHI, R; COUGHLAN, J. (Computer) vision without sight. Communications of the ACM, vol. 55, nº 1, p. 96–104, 2012.
9. ONU. World Report on vision. Disponível em: <https://www.who.int/publications/i/item/world-report-on-vision>. Acessado em 05/08/2022. Organização das Nações Unidas, Genebra, Suíça, 2019.
10. OQUAB, M.; BOTTOU, L.; LAPTEV, I.; SIVIC, J. Learning and transferring mid-level image representative using convolutional neural networks. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June, 2014.
11. PEDREGOSA, F. et al. Scikit-learn: machine learning in Python. Journal of Machine Learning Research, v. 12, p. 2825-2830, 2011.
12. PINEDA, A. L. et al. Comparison of machine learning classifiers for influenza detection from emergency department free-text reports. J Biomed Inform., Elsevier, v. 58, p. 60-69, 2015.
13. RUSSAKOVSKY, O. et al. ImageNet large scale visual recognition challenge. International Journal of Computer Vision (IJCV), v. 115, n. 3, p. 211-252, Dec 2015.
14. SCHMIDHUBER, J. Deep learning in neural networks: an overview. Neural Networks, v. 61, p. 85-117, January 2015.
15. SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. Computational and Biological Learning Society, v. 6, p. 1-14, 10 Apr 2015.