

# Sistema Embarcado Linux para Reconhecimento de Vogais em LIBRAS Utilizando Aprendizagem de Máquina

Gabriel S. Cavalcante<sup>1</sup>, Paulo M. M. da Silva<sup>1</sup>, Sandro C. S. Jucá<sup>1</sup>

<sup>1</sup>Departamento de Informática, Instituto Federal do Ceará  
CEP 61.919-140 – Maracanaú – CE – Brasil

{gabrielsoucavalcante, mateuu, sandro.juca}@gmail.com

**Abstract.** *The purpose of this paper is to build an application that initially recognizes vowel signals to help deaf people to maintain conversations through a communication for lay people in the Brazilian Sign Language. Audience delimitation was chosen because it is necessary to reduce communication barriers due to the lack of general knowledge in sign language. Through the use of a Raspberry Pi configured with TensorFlow and OpenCV for vowel signal recognition and the ability to send messages to Telegram.*

**Resumo.** *O objetivo deste trabalho é construir uma aplicação que reconheça inicialmente sinais de vogais para ajudar pessoas surdas, surdas-mudas e deficientes auditivas a manter conversas por um meio de comunicação para leigos na Língua Brasileira de Sinais. A delimitação do público foi escolhida porque é necessário reduzir as barreiras de comunicação pela falta de conhecimento geral em linguagem de sinais. Através do uso de um Raspberry Pi configurado com o TensorFlow e OpenCV para reconhecimento de sinais de vogais e com a capacidade de enviar mensagens para o Telegram.*

## 1. Introdução

Através da evolução computacional ocorrida nas últimas décadas, a comunicação humana e as formas de interações foram atualizadas pelas facilidades introduzidas pelas novas tecnologias de comunicação. Avanços mais recentes permitem ainda uma comunicação entre falantes de línguas distintas através de traduções simultâneas.

No entanto, o reconhecimento automático e a tradução de línguas gestuais ainda está em um estado inicial de desenvolvimento [Digiampietri et al. 2012]. Para que haja uma comunicação entre um falante de uma língua visual como a Língua Brasileira de Sinais (LIBRAS) e uma língua oral-auditiva como Língua Portuguesa, é necessário uma interface de comunicação em que há *i*) o reconhecimento automático dos sinais e sua conversão para português escrito ou falado e *ii*) o reconhecimento de voz e sua tradução visual [Digiampietri et al. 2012]. Para o segundo ponto, muitos aplicativos atuais trabalham com o reconhecimento de voz para converter o idioma falado em forma de texto [Dietz 2002] e de tradução visual [Corrêa et al. 2014].

Este trabalho busca estudar formas de facilitar a traduções simultâneas de LIBRAS para Português Brasileiro através de uma solução para reconhecimento automático dos gestos de vogais utilizando o Sistema Embarcado Linux instalado em um Raspberry Pi com os sinais capturados por uma câmera e identificados por um modelo de aprendizagem de máquina para detecção automática de objetos. O modelo aplicado foi o Single Shot

MultiBox Detector (SSD), treinado com um novo conjunto de imagens composto das fotografias dos gestos, seus rótulos e as caixas de contorno do local na imagem.

Dado que esta tecnologia assistiva possui o potencial de ser aplicada em diversos ambientes, como salas de aula, áreas de atendimento ao público e residências, neste trabalho integramos a ferramenta com o aplicativo para troca de mensagens Telegram.

## 2. Fundamentação Teórica

Este projeto consiste em reconhecer padrões correspondentes a vogais em LIBRAS. Para o processamento, foi utilizado o Raspberry Pi 3 B+, um Sistema Embarcado Linux de baixo custo e de fácil acesso. Suas principais características consistem em um processador Broadcom BCM2837 que é eficiente na dissipação de calor, tornando-o capaz de operar em 1.4GHz, facilitando o processamento contínuo em tempo real. Possui memória RAM de 1GB LPDDR2 e a capacidade de armazenamento provém de um cartão MicroSD [Bauermeister 2018]. Também foi utilizado a Câmera Pi por capturar imagens com até 2592x1944 *pixels* resolução e vídeos com até 1080p de resolução.

Afim de somar ao estudos relacionados a reconhecimento de sinais e facilitar futuras implementações, foi escolhido a biblioteca de software de código aberto TensorFlow. Esta biblioteca é multiplataforma e pode ser utilizado em sistemas operacionais como Windows, Linux e MacOS. A linguagem utilizada pela biblioteca é Python 3.7.

## 3. Pesquisa Documental

O estudo na área de reconhecimento automático de sinais de LIBRAS teve um grande avanço nos últimos anos. Pode-se encontrar diferentes formas de atingir bons resultados, como demonstrado nos trabalhos que utilizam câmera de profundidade [Santos et al. 2015, de Melo Gois 2014] e outros que utilizam a classificação das imagens em computadores de propósito geral [Digiampietri et al. 2012, Barros et al. 2014, Gonçalves and Alves 2010]. No entanto, não foram encontrados trabalhos envolvendo a aplicação de sistemas embarcados para o tratamento do problema.

Desta forma, a pesquisa documental estendeu-se para a identificação de trabalhos que utilizam sistemas embarcados em tarefas de processamento e reconhecimento de imagens. Nos artigos selecionados, todos utilizam um Raspberry Pi em conjunto com a Câmera Pi. A Tabela 1 de comparação apresenta as características em comum dos trabalhos. Nota-se que existe uma grande variedade de algoritmos foram utilizados para as tarefas de reconhecimento de padrões com Raspberry Pi (RPi), incluindo processamento de detecção de objetos ligados a internet na nuvem.

**Tabela 1. Comparação entre projetos para processamento e reconhecimento de imagens com sistemas embarcados**

<b>Trabalho</b>	<b>Modelo</b>	<b>Algoritmo de Classificação</b>	<b>Internet</b>
[Dürr et al. 2015]	RPi B	CNN	Não
[Hadidi et al. 2018]	RPi 3B	AlexNet e VGG16	Não
[Patel and Verma 2017]	RPi 3B	IBM Watson	Sim
[Buhus et al. 2016]	RPi 3B	OpenALPR	Não

## 4. Metodologia

A identificação dos diferentes símbolos foi feito pelo modelo de aprendizagem profunda Single Shot MultiBox Detector (SSD) [Liu et al. 2016]. A ideia consiste em treinar diferentes símbolos que serão reconhecidos na imagem. Como o modelo necessita de que os objetos/símbolos estejam devidamente marcados com caixas de contorno nas imagens, foi necessário construir uma nova base de dados, pois não foi encontrada uma base adequada.

Figura 1 apresenta um diagrama com os passos da metodologia aplicada: *i*) Um banco de imagens com fotografias dos símbolos foi criado, *ii*) cada imagem foi demarcada com uma caixa limitadora indicando a posição do símbolo na imagem, *iii*) o modelo SSD foi treinado com a base de dados em um computador pessoal. Estando o modelo treinado, este é configurado no Raspberry Pi. Para a classificação, *v*) as imagens são capturadas, *v*) os símbolos são identificados e traduzidos nas vogais e, por fim, *vi*) enviadas pelo aplicativo de mensagens Telegram.

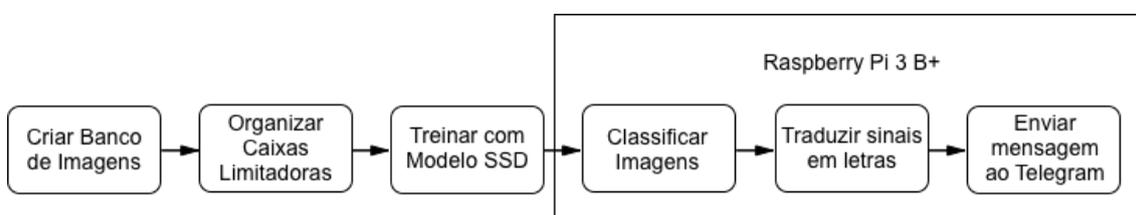


Figura 1. Metodologia aplicada no projeto

### 4.1. Banco de Imagens

O banco de imagens construído é formado pelas capturas das vogais (A, E, I, O, U) e também de um sinal para o envio de mensagem para o aplicativo Telegram. A quantidade capturada para cada sinal é de 40, somando um total de 240 imagens. Dado que o modelo SSD necessita da informação de onde os objetos se encontram na imagem, foi criado um arquivo de formato `xml` contendo a definição das coordenadas da caixa delimitadora para cada imagem.

Nesta versão inicial do trabalho, foi escolhido trabalhar somente com vogais devido ao grande número de amostras de imagens que teriam que ser recolhidas em caso de reconhecimento do alfabeto por completo, mas existe a possibilidade de ser feito em trabalhos futuros.

### 4.2. Treinamento do modelo

SSD é um modelo leve para detecção de objetos em imagens usando uma rede neural profunda. O mesmo discretiza o espaço de saída das caixas delimitadoras em um conjunto de caixas padrão sobre diferentes proporções e escalas por localização de mapa [Liu et al. 2016]. Entretanto, o custo computacional para o treinamento é alto e por isso foi utilizado um Macbook Pro com processador 2,2 GHz Intel Core i7 (I7-8750H).

Para a realização do treinamento, as imagens correspondentes a um mesmo sinal foram aleatoriamente separadas em dois grupos, treinamento com 80% das imagens e teste com 20%. O processo de aprendizagem teve duração de aproximadamente 16 horas, ocorrendo um pouco mais de 10 mil iterações como demonstrado na figura 2. Quando o modelo está treinado, basta passá-lo para o Raspberry Pi.

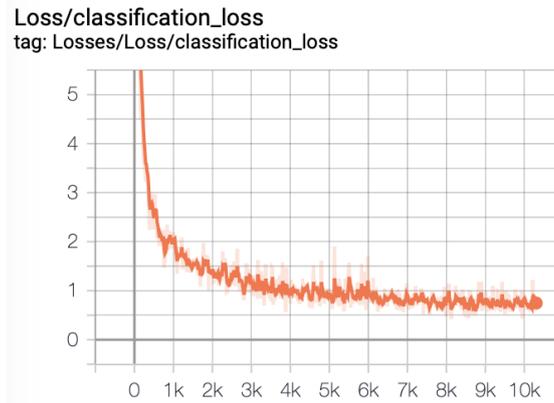


Figura 2. Gráfico de perdas de treinamento no TensorFlow

### 4.3. Identificação dos Sinais

Estando o Raspberry Pi devidamente configurado com modelo, a identificação dos sinais em tempo real ocorre quando o vídeo da Câmera Pi é capturado quadro-a-quadro por meio da biblioteca OpenCV [Bradski 2000] um quadro por vez é passado para o modelo SSD. O modelo treinado atribui à imagem do quadro uma caixa de contorno para cada possível objeto detectado e associa a cada caixa uma taxa de confiança a caixa delimitadora com maior taxa de confiança é selecionada. O objeto (i.e., a classe) associado à caixa é retornado.

### 4.4. Telegram Bot

A solução desenvolvida foi integrada com o Telegram por meio de um *bot*, no qual vogais reconhecidas pelo sistema foram enviadas. Para isso os quadros são classificados com o símbolo de maior taxa de confiança. Caso a taxa de confiança seja igual ou superior um limiar estabelecido, o caractere é concatenado a uma cadeia de caracteres (*string*) mensagem. Quando o modelo reconhece o símbolo correspondente ao de envio de mensagem, a mensagem é encaminhada para o bot. O sinal definido para o envio de mensagem é o indicado na Figura 3-a o bot recebe a mensagem, envia ela para o Telegram e deixa a cadeia de caracteres mensagem vazia, e a mensagem chega para o contato (Figura 3-b).



(a) Reconhecimento de sinal de enviar mensagem



(b) Recepção de mensagens no aplicativo Telegram

Figura 3. Funcionalidade de envio de mensagens ao Telegram

## 5. Experimentos computacionais

Para validação de que os modelos estavam devidamente treinados, imagens de teste separadas antes do treinamento foram classificadas com uma taxa de acerto de 96,67%. Na maioria dos erros, o modelo não indicou uma confiança significativa. Um possível tratamento para estes casos seria a consideração da classificação do quadro anterior ou do quadro posterior.

Os testes possuem algumas variâncias devido a dificuldade no reconhecimento de sinais através das mãos, já que possuem aspectos visuais parecidos. A Figura 4 demonstra o resultado da classificação do sinal da letra U com variação para o sinal da letra I. Foi visto também que o resultado dos sinais encontrados são afetados pela luminosidade e distância. Além disso Câmera Pi possui uma variação de quadros por segundo entre 11 e 17, que muitos deles passam para o algoritmo desfocado.

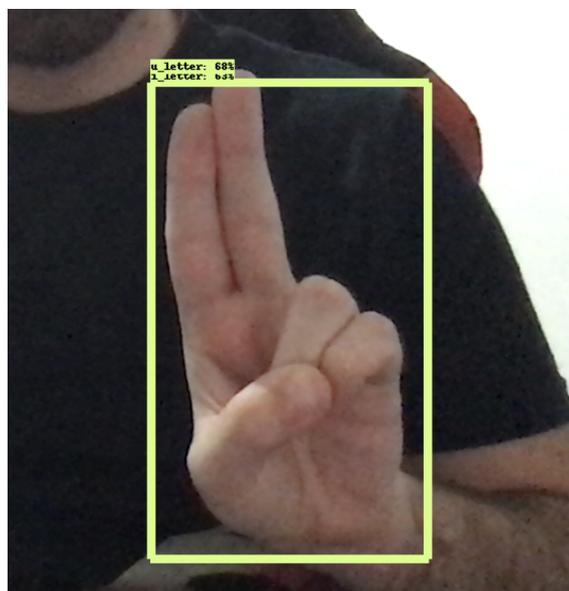


Figura 4. Classificação do sinal da letra U

## 6. Conclusão

Os resultados demonstrados nesse artigo são satisfatórios, porém as classificações de imagens em tempo real não são tão assertivas. Para evitar isso, seria necessário uma maior base de imagens com diferentes ambientes e iluminações. Outra forma que pode ser feita em trabalhos futuros é trabalhar com a entrada de até 5 quadros e selecionar por eleição, ou seja, fazer com que o resultado da seleção seja a maioria identificada. Desta forma, provavelmente a identificação de sinais em tempo real seriam mais assertivas.

Um dos pontos positivos foi a facilidade de utilizar o Raspberry Pi, sendo um computador completo e de alto poder de processamento para diversas resoluções de problemas. O envio de mensagens para o Telegram é também uma forma de demonstrar as diversas características e funcionalidades pode-se realizar simultaneamente.

Pôde-se entender que existem diversas formas de tornar a acessibilidade parte do dia a dia de pessoas portadoras de deficiências através de visão computacional. Através

deste estudo e objetivos alcançados, foram abertas algumas portas para futuras aplicações de acessibilidade.

## Referências

- Barros, R. A., Pontes, A. V., and Almeida, J. D. (2014). Reconhecimento de linguagem de sinais: aplicação em libras. *JIM 2014-V Jornada de Informática do Maranhão*.
- Bauermeister, G. (2018). Nova raspberry pi 3 b+. <https://blog.fazedores.com/nova-raspberry-pi-3-b-plus/>.
- Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.
- Buhus, E. R., Timis, D., and Apatean, A. (2016). Automatic parking access using openalpr on raspberry pi3. *Acta Technica Napocensis*, 57(3):10.
- Corrêa, Y., Vieira, M. C., Santarosa, L. M. C., and Biasuz, M. C. V. (2014). Aplicativos de tradução para libras e a busca pela validade social da tecnologia assistiva. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 25, page 164.
- de Melo Gois, G. B. (2014). Reconhecimento do alfabeto de libras usando sensor kinect e marcadores visuais.
- Dietz, T. A. (2002). Speech recognition text-based language conversion and text-to-speech in a client-server configuration to enable language translation devices. US Patent 6,385,586.
- Digiampietri, L. A., Teodoro, B., Santiago, C. R., Oliveira, G. A., and Araujo, J. C. (2012). Um sistema de informação extensível para o reconhecimento automático de libras. *SBSI-Simpósio Brasileiro de Sistemas de Informação*.
- Dürr, O., Pauchard, Y., Browarnik, D., Axthelm, R., and Loeser, M. (2015). Deep learning on a raspberry pi for real time face recognition. In *Eurographics (Posters)*, pages 11–12.
- Gonçalves, D. P. and Alves, A. G. (2010). Ferramenta para aprendizado de libras para crianças do ensino fundamental utilizando visão computacional.
- Hadidi, R., Cao, J., Woodward, M., Ryoo, M. S., and Kim, H. (2018). Real-time image recognition using collaborative iot devices. In *Proceedings of the 1st on Reproducible Quality-Efficient Systems Tournament on Co-designing Pareto-efficient Deep Learning*, page 4. ACM.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer.
- Patel, A. and Verma, A. (2017). Iot based facial recognition door access control home security system. *International Journal of Computer Applications*, 172(7):11–17.
- Santos, J. R. d. et al. (2015). Reconhecimento das configurações de mão de libras baseado na análise de discriminante de fisher bidimensional utilizando imagens de profundidade.