



Tradução de Libras por Imagem

Daniel Rodrigo Santos Ribeiro¹

Marco Túlio Alves Nolasco Rodrigues²

Resumo: A Língua Brasileira de Sinais (Libras), é uma das formas de comunicação entre as comunidades de pessoas surdas ou que apresente alguma necessidade especial auditiva. Esta linguagem é formada por sua estrutura sintática, semântica e morfológica. Aqueles que não são conhecedores da língua de sinais, podem utilizar de aplicativos que traduzem a Libras para a linguagem portuguesa, mas há uma carência de aplicações que fazem a tradução inversa da língua de sinais para a língua portuguesa. Devido a complexidade dos gestos da linguagem de sinais que utilizam mãos, rostos, boca enfim todo o corpo, impacta na criação de aplicativos com fim de tradução da linguagem de sinais para língua portuguesa. Há também uma carência de datasets com os sinais da Libras Brasileira para estudo voltado a finalidade de reconhecimento e interpretação destes sinais. O mapeamento destes sinais e a criação de um dataset é de grande importância para consolidar o desenvolvimento de aplicações para tradução de Libras para língua portuguesa. Com o avanço na área de visão computacional, a cada dia, proporciona meios de desenvolvimento de aplicações mais complexas, permitindo a interação homem máquina de modo mais simples. Este trabalho propõe a criação de um dataset de imagens para o uso em aplicação que traduza a linguagem de sinais para a língua portuguesa, assim concedendo mais liberdade a comunidade de deficientes auditivos, tornando a comunicação bilateral.

Palavras-chave: Libras, Visão Computacional, Comunicação.

^{1.} Ciência da Computação, Graduado, Universidade de Itaúna, drsantosribeiro@gmail.com.

^{2.} Ciência da Computação, Doutorado, Universidade de Itaúna, tulio.rodrigues@gmail.com.

1. Introdução

Dados do IBGE mostram que cerca de 9 milhões de pessoas no Brasil são portadores de alguma necessidade especial auditiva. Dentre esse número existe a comunidade surda que é dependente da Língua Brasileira de Libras Sinais (Libras) [QUADROS and Cruz 2009] para realizar a comunicação entre eles e os ouvintes. Algumas ferramentas são importantes para proporcionar a interação com um portador de necessidade especial auditiva como *Hand talk* [Talk 2016] ou *VLibras* [VLibras 2016], ambos fazem a tradução da língua portuguesa para a língua materna dos surdos. Existe uma carência em aplicações que fazem a tradução inversa, deixando assim a comunidade surda em uma situação passiva em nossa sociedade. É grande a falta de dataset encontrada para utilização em estudos, que visam a criação de ferramentas que traduzam os sinais da Libras para língua portuguesa. Uma pequena parcela da população é dotada do conhecimento da língua de sinais ou de ferramentas que auxiliam na interação com um portador de necessidade especial auditiva. Os gestos da linguagem de sinais representam letras do alfabeto como demonstra a FIGURA 1, e também é composta por sinais que representam palavras, verbos, cores, formas, objetos sendo uma linguagem estruturada.

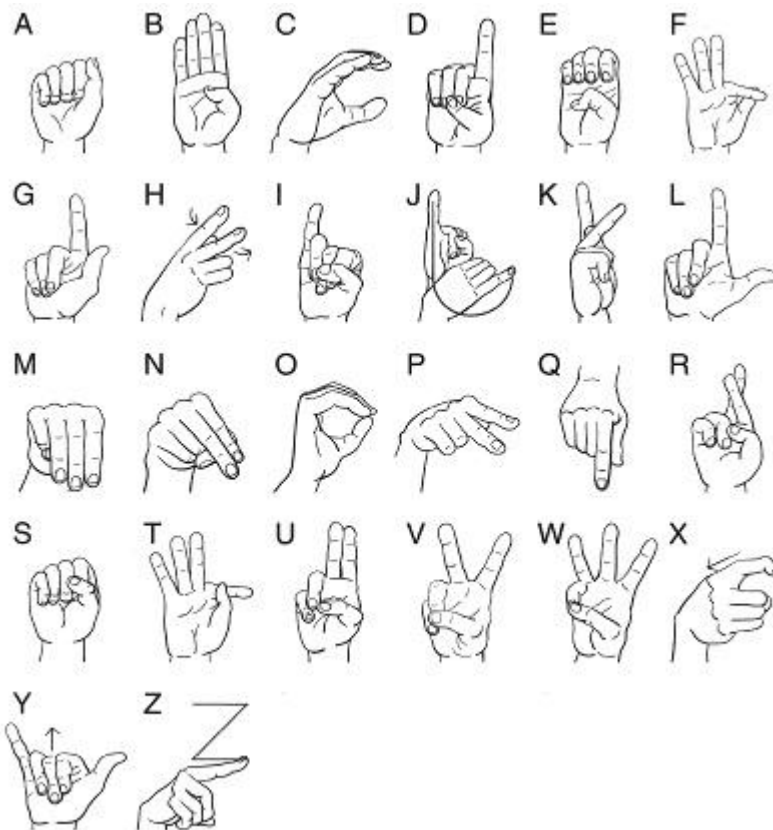


FIGURA 1. Representação do alfabeto português ortografado em Libras.

A proposta deste trabalho é o mapeamento das letras do alfabeto em libras, para a criação de um *dataset* de imagens. Desenvolvendo uma base de dados sólida contendo não somente imagens das letras do alfabeto, mas também os sinais que correspondem a palavras, auxiliaria o estudo e desenvolvimento de aplicações voltadas para a identificação e tradução dos sinais da Libras. Utilizando a biblioteca *Opencv* (*Open Source Computer Vision Library*) e seus classificadores, mapeamos os sinais que expressam as letras do alfabeto onde o desenvolvimento de uma aplicação é capaz de receber uma imagem através de uma câmera estacionária devidamente calibrada, identificar o sinal captado e traduzir para a língua portuguesa. O desenvolvimento de uma aplicação para esta finalidade utiliza classificadores gerados a partir do dataset como mostra a FIGURA 2 onde cada imagem representa uma letra do alfabeto definida pela linguagem de sinais.



FIGURA 2. Letras do alfabeto executado em Libras. Sinais do alfabeto de A a Z sendo fotografados para criação do *dataset*.

Identificado estes sinais e utilizando uma base de dados devidamente treinada, após a captura do sinal por uma câmera estacionária é realizado o reconhecimento do sinal, a tradução para o português e apresentado em forma de texto para o usuário, criando assim um canal de comunicação para aqueles que não são conhecedores da Libras, possam interagir com um portador de necessidades especiais auditivas. A aplicação desenvolvida consiste em receber uma imagem de uma *webcam* e realizar o processamento comparando com os classificadores gerados pelo *dataset* criado.

2. Metodologia

Nesta seção, é descrita a metodologia utilizada para criação do *dataset*, a construção do algoritmo, a cascata de classificadores, insucesso na utilização da biblioteca *Canny Edge*

Detector que identifica contornos, e as bibliotecas utilizadas da *OpenCV* (*Open Source Computer Vision Library*).

2.1 Criação do *Dataset*

A baixa disponibilidade de datasets no Brasil contendo os sinais da Libras, um *dataset* de língua de sinais de outra nacionalidade como a língua de sinais americana denominada de *ASL* (*America Sign Language*) não pode ser utilizada para o nosso país devido algumas letras e palavras serem representadas de modo diferente. A FIGURA 3 (a) mostra sinal Q em libras com a mão fechada e indicador para baixo com polegar ao lado, e (b) apresentam letra Q por meio de Libras e *ASL* demonstrando a diferença na execução do sinal.

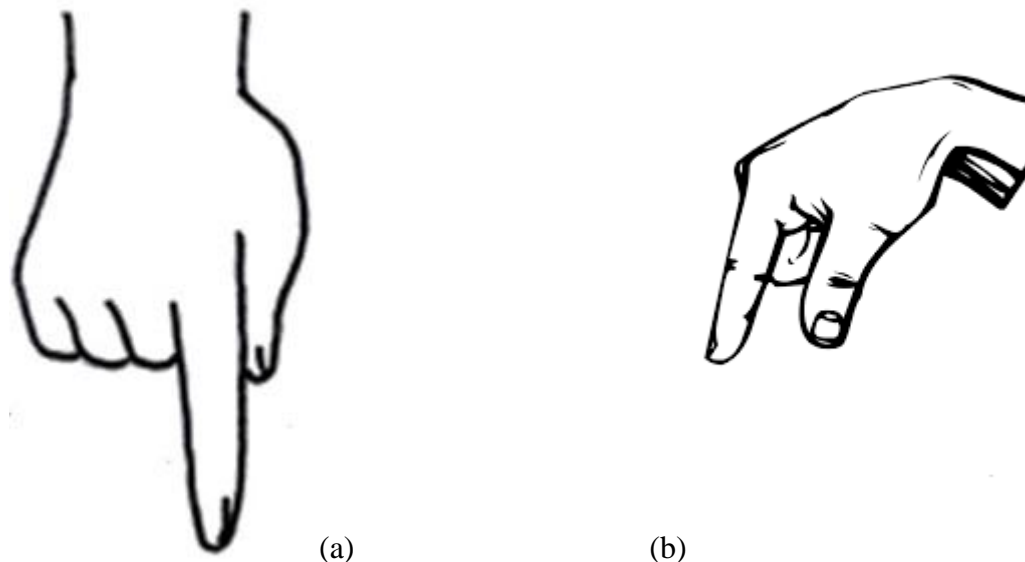


FIGURA 3. Representação de sinais. Em (a) apresenta a representação da letra Q em Libras. Já em (b) apresenta a representação da Letra Q em *ASL*.

Para a criação do *dataset* realizamos a captação da imagem da mão direita dos usuários emitindo os sinais do alfabeto em Libras utilizando uma câmera digital. O *dataset* é dividido entre sinais positivos e negativos, cada letra é executada de maneira correta e errada. Os sinais positivos de cada letra foram fotografados 10 vezes em profundidades diferentes, sinais negativos fotografados não tiveram sua profundidade alternada, sendo estas imagens convertidas para cinza.

2.2 Algoritmo Desenvolvido

A implementação do algoritmo é desenvolvida utilizando a linguagem de programação *Python*, o código consiste em abrir uma webcam, criação de uma área de interesse criando uma janela menor obtendo ganho em processamento, realizar a correção de ruídos e



conversão para cinza do vídeo recebido. A imagem é analisada e quando identificada emite a letra correspondente em tela. Abaixo observa-se parte do código implementado.

```
1 import numpy as np
2 import cv2
3
4     Acascade=cv2.CascadeClassifier('C:\\opencv\\sources\\data\\
5     haarcascades\\AA.xml')
6
7     captura=cv2.VideoCapture(0)#ABERTURA DE CAMERA ^
8
9     captura.set(3,640)
10    captura.set(4,480)
11
12    while(True):
13
14        ret,frame=captura.read()#LEITURA DO FRAME
15
16        roi=frame[80:440,130:490]#CRIA REGIAO~ DE INTERESSE
17    kernel=np.ones((5,5),np.uint8)
18        roi=cv2.morphologyEx(roi,cv2.MORPH_OPEN,kernel)
19
20        gray=cv2.cvtColor(roi,cv2.COLOR_BGR2GRAY)#CONVERTE
21        PARA TOM DE CINZA
22
23        sinalA=Acascade.detectMultiScale(gray,scaleFactor=1.05,
24        minNeighbors=2,minSize=(51,47),maxSize=(106,87),flags=cv2.
25        CASCADE_SCALE_IMAGE)
26        for(x,y,w,h)insinalA:
27            printx,y,w,h
28            font=cv2.FONT_HERSHEY_SIMPLEX
29
30            cv2.putText(frame,'A',(60+109,39+80),font,5.0,(11,255,255),2,
31            cv2.LINE_AA)
32
33            cv2.imshow('ROI',roi)
34            cv2.imshow('Frame',frame)
35
36            ifcv2.waitKey(1)&0xFF==ord('q'):
37                break
38
39    cap.release()
40    cv2.destroyAllWindows()
```

A escolha da linguagem de programação *Python* ocorreu por se tratar de uma linguagem de fácil implementação e que já está bem integrada a biblioteca *OpenCV*. O ganho em



produtividade entre Python e a linguagem C++ muito utilizada para trabalhar com a OpenCV é considerável.

2.3 Cascata de Classificadores

OpenCV dispõe da cascata de classificadores [Viola and Jones 2001] biblioteca importante de detecção de faces e objetos. Cascata de classificadores é uma abordagem de aprendizado de máquina, que uma função em cascata é treinada utilizando várias imagens positivas e negativas. A biblioteca analisa as amostras positivas criando padrões entre as imagens negativas em tom de cinza criando padrões de diferenças de intensidade em regiões distintas. Após análise destes padrões ela armazena as características em uma árvore de decisões. A FIGURA 4 mostra parte de um código de cascata de classificadores que são analisados em sequência para criação deste padrão. Para os classificadores executarem um bom treino é necessário uma grande quantidade de imagens positivas e negativas em posições e situações diferentes, onde será associado aos classificadores as diferenças encontradas. É importante que o número de imagens para criação de um dataset seja feito com várias fotos dos sinais em ângulos diferentes, para que seja feito o reconhecimento mais preciso e imediato. As amostras negativas também são um ponto de extrema importância para criação de uma base robusta, onde o cuidado em criar situações adversas para mapear gestos que não representam um sinal da linguagem de libras.

```
<?xml version="1.0"?>
<opencv_storage>
<myhaar type_id="opencv-haar-classifier">
  <size>
    24 24</size>
  <stages>
    <_>
      <trees>
        <_>
          <_>
            <feature>
              <rects>
                <_>
                  0 0 16 2 -1.</_>
                <_>
                  0 1 16 1 2.</_></rects>
              <tilted>0</tilted></feature>
              <threshold>-9.6788266673684120e-003</threshold>
              <left_val>1.</left_val>
              <right_val>-1.0000020265579224</right_val></_></_></trees>
            <stage_threshold>1.</stage_threshold>
```

FIGURA 4. Imagem mostra parte do código gerado pela cascata de classificadores.



2.4 Biblioteca *Canny Edge*

Canny Edge é um poderoso algoritmo de detecção de bordas composto por vários estágios. A detecção de bordas é suscetível ao ruído da imagem. No primeiro estágio para esta correção o algoritmo remove o ruído utilizando o filtro gaussiano. Sendo a imagem suavizada, ela é filtrada com um núcleo de *Sobel* que encontra o gradiente e a direção da borda para cada pixel. Depois de encontrar o gradiente é feita uma varredura completa na imagem para remover os pixels indesejados que podem não constituir a borda. O último estágio é o limite de histerese que decide quais bordas são realmente bordas e quais não são. O insucesso na utilização deste algoritmo é devido ao não reconhecimento total das bordas da mão a FIGURA 5, como demonstrado na figura o que diferencia o sinal da letra K da letra V é o posicionamento do dedo polegar sobre o dedo indicador. É visto na figura que a detecção da borda do dedo polegar é quase imperceptível, causando erro na identificação do sinal K sendo interpretado como o sinal V.

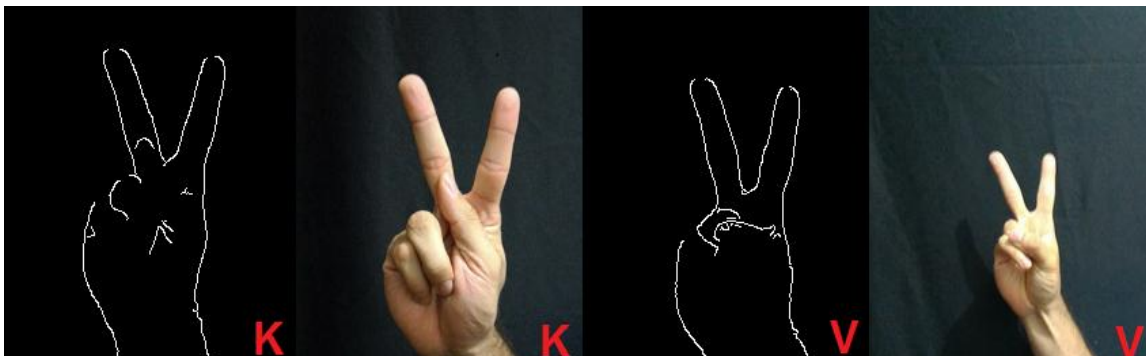


FIGURA 5. Semelhança entre os sinais da letra K e V. Ausência do contorno total do polegar na letra K. Letra K é identificada como letra V.

3. Resultados

Para demonstrar o uso de um *dataset* voltado para o desenvolvimento de uma aplicação que interprete os sinais da Libras, a aplicação desenvolvida foi apresentada aos usuários, que se submeteram ao teste de reconhecimento dos sinais da Libras. Entre os usuários o domínio da Libras por parte deles era de total, médio e nenhum conhecimento. Os usuários tiveram um tempo pré-estabelecido de 10 segundos onde um determinado sinal deveria ser lido e interpretado. As vinte e seis configurações da Libras, referente aos sinais do alfabeto, foram executadas dez vezes cada letra. Os sinais que ultrapassaram o tempo determinado para identificação não foram considerados como acerto. Para quantificar estes dados utilizamos a média geométrica $\sqrt[x]{a * b * c \dots}$ um histograma também exibe a frequência em que os sinais foram identificados.



3.1 Usuário 1

O usuário 1 apresentou um média geométrica $\sqrt[26]{1; 555492378 * 10^{18}} = 5; 00$. Este usuário tem uma bom conhecimento da língua de sinais. A FIGURA 6 demonstra o histograma do reconhecimento dos sinais. O eixo horizontal assinala as letras de A a Z, já o eixo vertical marca dentro do numero de tentativas conforme mencionado acima, a quantidade de vezes que um determinado sinal foi identificado. Pode-se observar que o usuário com maior domínio da linguagem obteve a maior média geométrica. Estes bons resultados foram alcançados pela facilidade e precisão na execução dos sinais. Um sinal com clareza na sua execução gera maiores informações positivas que são comparadas com os classificadores gerados pela base de dados criada.

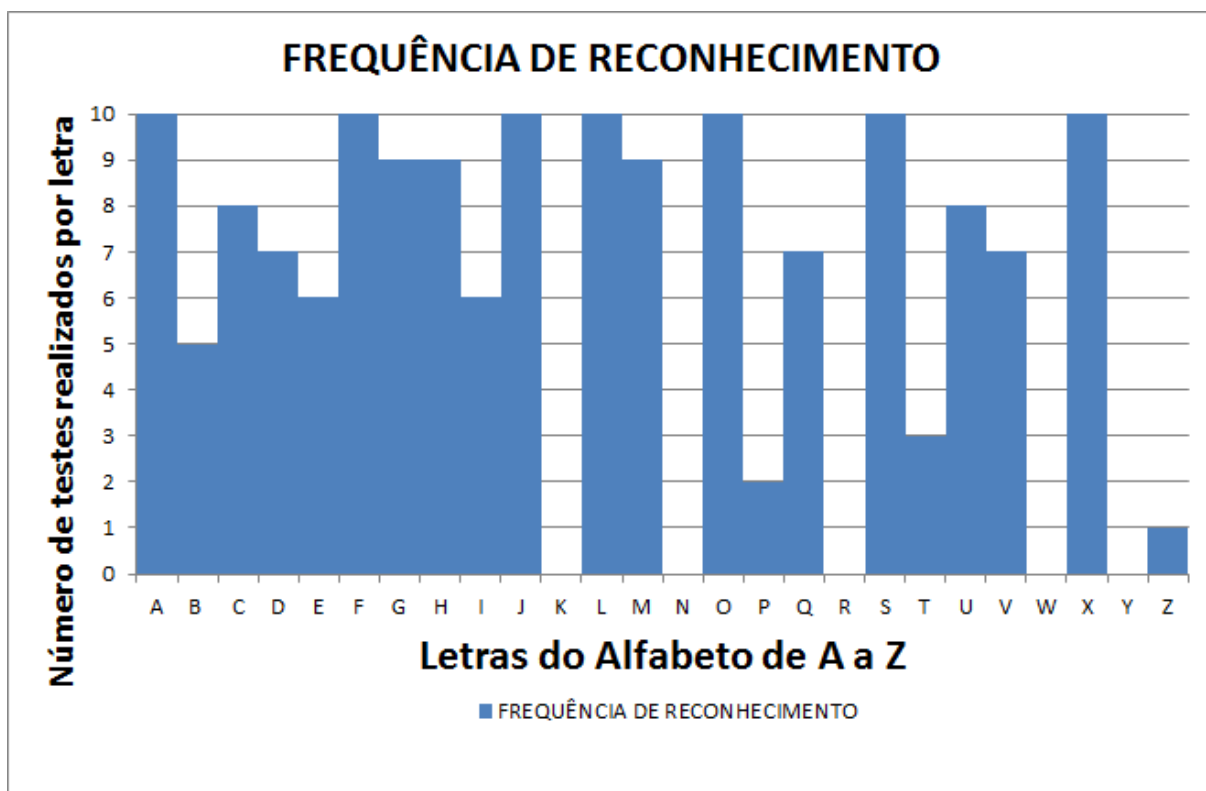


FIGURA 6. Histograma do teste de reconhecimento dos sinais usuário 1.

3.2 Usuário 2

O usuário 2 apresentou um média geométrica $\sqrt[26]{6.58409472 * 10^{14}} = 3,71$. O usuário é dotado de médio conhecimento da língua de sinais, a FIGURA 7 exibe seus resultados apurados. Observa-se que a média geométrica apresentada manteve muito próxima ao usuário 1, o fator importante foi o conhecimento da linguagem por parte deste usuário.

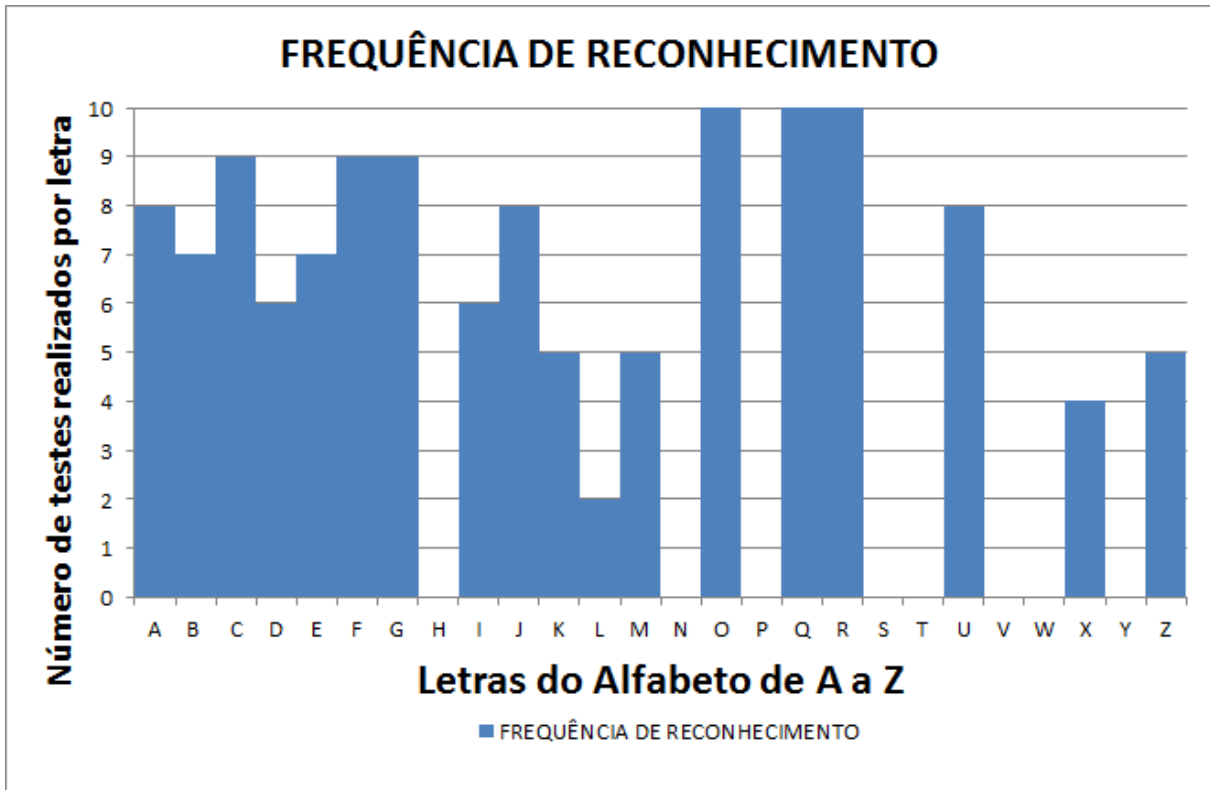


FIGURA 7. Histograma do teste de reconhecimento dos sinais usuário 2.

3.3 Usuário 3

O usuário 3 apresentou uma média geométrica $\sqrt[26]{4.72392 * 10^{11}} = 2,81$. Este usuário não era conhecedor da Libras, após a apresentação da linguagem o usuário foi submetido ao teste. Este usuário apresentou dificuldade na execução dos sinais alcançando uma média geométrica inferior aos demais usuários como mostra a FIGURA 8 que relaciona seus resultados. Mesmo apresentando uma média inferior aos demais foi comprovada que basta um pouco de conhecimento na linguagem de sinais, que um interpretador auxiliaria a comunicação deste usuário.

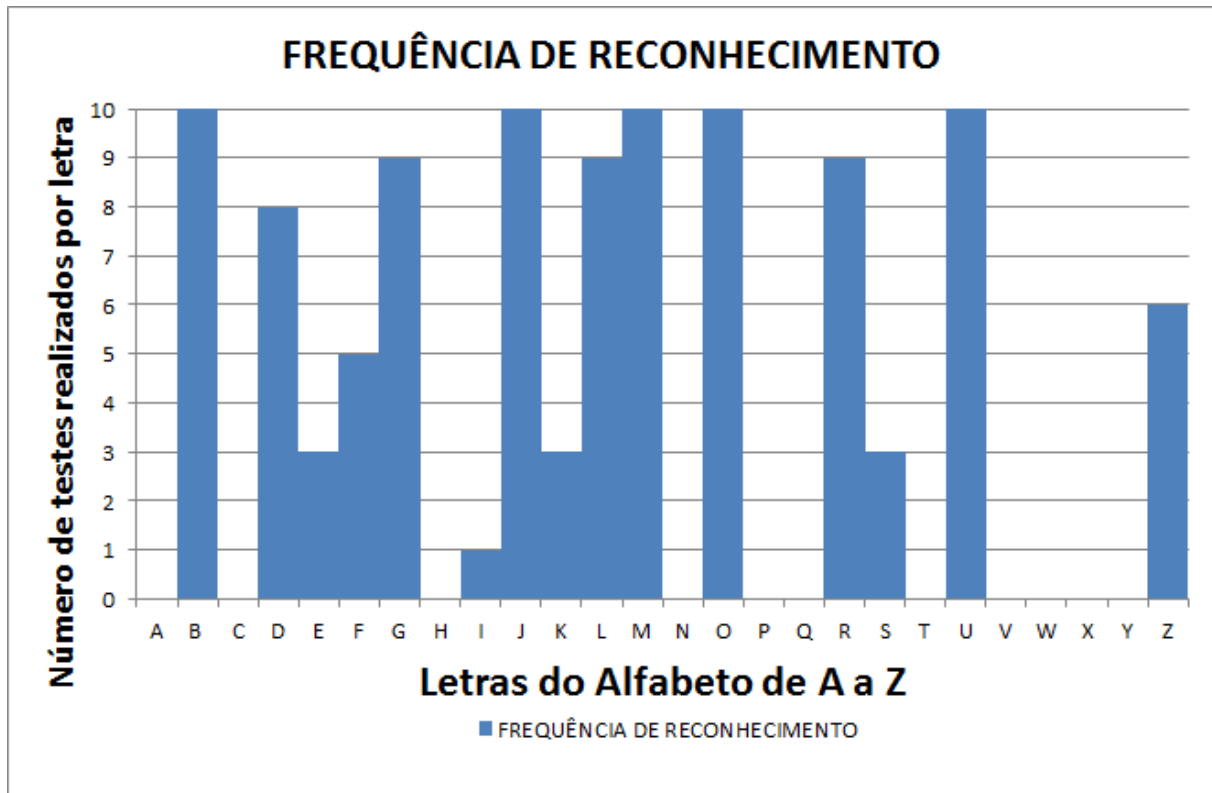


FIGURA 8. Histograma do teste de reconhecimento dos sinais usuário 3.

4. Conclusão

A Linguagem Brasileira de Sinais é muito bem estruturada, onde pode se classificar e mapear suas características. Extraíndo suas características e definindo uma base de dados sólida, podemos utilizá-la na criação de ferramentas para auxiliar na comunicação de um portador de necessidade especial auditiva ou aprendizado da língua de sinais por um ouvinte. Pontos importantes foram observados na criação do dataset e no uso da aplicação desenvolvida. Aspectos como iluminação do ambiente para captação dos sinais para o *dataset* e leitura dos sinais através de uma *webcam* pelo interpretador são fatores de grande influência que impacta na identificação dos sinais. Temos que observar também as semelhanças de alguns sinais que são executados quase que da mesma maneira. Verificamos também que os usuários com maior conhecimento dos sinais da Libras, executam os sinais de modo mais natural facilitando assim a leitura e interpretação dos sinais. Os testes para o mapeamento do alfabeto apresentou bons resultados na identificação dos sinais realizados. Com o uso de um *dataset* bem elaborado para linguagem de sinais, e o avanço crescente no campo da visão computacional é viável a utilização de um *dataset* para o estudo e desenvolvimento de aplicações voltadas a interpretação e tradução da linguagem de sinais para língua portuguesa falada e escrita.

Referências



SICIT
Semana de Iniciação
Científica e Tecnológica

25 a 29 de setembro de 2017
Engenharias e Computação

 Universidade de Itaúna

DE QUADROS, Ronice Müller; KARNOPP, Lodenir Becker. **Língua de sinais brasileira: estudos lingüísticos**. Artmed Editora, 2009.

TALK, H. (2016). Hand Talk tradutor automatico para libras, a língua brasileira de sinais.

VIOLA, Paul; JONES, Michael J. Robust real-time face detection. **International journal of computer vision**, v. 57, n. 2, p. 137-154, 2004.

VLIBRAS (2016). Vlibras tradutor automatico para libras.