

# COMPUTER VISION AND NEURAL NETWORKS FOR LIBRAS RECOGNITION

Silas Luiz Furtado<sup>1</sup>, Jauvane de Oliveira<sup>2</sup>

<sup>1</sup> Instituto Militar de engenharia - SE/8 - silasfurtado@outlook.com.br

<sup>2</sup> Mathematical and computational methods department, National Laboratory for Scientific Computing - LNCC - jauvane@acm.org  
Rio de janeiro, Brasil

**Abstract**—In recent years, one can find several efforts to increase the inclusion of people with some type of disability. As a result, the global study of sign language has become an important research area. Therefore, this project aims at developing an information system for the automatic recognition of the Brazilian Sign Language (LIBRAS). The recognition shall be done through the processing of videos, without relying on support hardware. Given the great difficulty of creating a system for this purpose, an approach was developed by dividing the process into stages. In addition to dynamically identifying signs and context, neural network concepts and tools were used to extract the characteristics of interest and classify them accordingly. In addition, a dataset of signs, referring to the alphabet in LIBRAS, was built as well as a tool to interpret, with the aid of a webcam, the signal executed by a user, transcribing it on the screen.

**Index Terms**—LIBRAS, inclusion, disabilities, computer vision.

## I. INTRODUÇÃO

Não podemos ignorar que ao nosso redor, todos os dias milhares de pessoas enfrentam diversos obstáculos devido a deficiências de diversas naturezas. Dentre todas essas naturezas, podemos destacar a deficiência auditiva, quando o seu portador possui uma severa dificuldade em se comunicar em uma sociedade cuja comunicação se faz predominantemente por meios sonoros.

Dentre as deficiências existentes, de acordo com a organização mundial de saúde [1] cerca de 5% da população mundial, ou seja, por volta de 466 milhões de pessoas no mundo, possuem surdez incapacitante. Isso significa que apenas essa parcela de 5% possui perda auditiva maior que 40dB quando falamos do indivíduo adulto e 30dB nas crianças, englobando tipos de suave a severo, e que a estimativa é de que até o ano de 2050 este número aumente para cerca de 10% da população.

Olhando os números da realidade brasileira, segundo o último censo realizado em 2010 pelo IBGE [2] cerca de 10 milhões de brasileiros possuem deficiência auditiva, representando cerca de 5,1% da população total brasileira. Deste montante, cerca de 2 milhões de pessoas possuem deficiência severa considerando 344,2 mil não escutam por completo e 1,7 milhões têm grande dificuldade para escutar.

Para a maioria dessas pessoas, é de extrema dificuldade utilizar a língua portuguesa devido a não conseguir assimilar e replicar a oralidade da língua. Com isso é necessário que eles

se utilizem de uma língua visual-espacial para se comunicar, sendo LIBRAS a principal língua utilizada por deficientes auditivos.

Uma grande dificuldade no reconhecimento de LIBRAS é a quantidade de parâmetros que são considerados para que um sinal possua algum significado. De acordo com Stokoe [3] um sinal com algum significado é formado combinando cinco parâmetros: (a) configuração de mão; (b) ponto de articulação; (c) movimento; (d) orientação/direção; e (e) expressões não manuais.

Além disso, apesar de existir configurações de mão representando todas as letras do alfabeto, na comunicação diária não é comum serem utilizados essas configurações e sim gestos referentes a verbetes para formação de sentenças. Nesse caso, na segmentação das imagens para que um gesto seja reconhecido é necessário uma grande capacidade computacional.

Este trabalho inspira-se em trabalhos prévios [4] e, através do uso de visão computacional, aplica uma metodologia e a mostra através de um protótipo capaz de, com o auxílio de uma *webcam*, analisar e reconhecer sinais do alfabeto de LIBRAS, traduzindo automaticamente para a língua portuguesa e transcrevendo interativamente seu significado na tela do computador.

## II. METODOLOGIA PARA O SISTEMA E CONCEITOS

Para facilitar o entendimento, serão apresentadas as etapas principais do sistema de maneira modular. Cada etapa da metodologia adotada corresponde a um módulo do sistema, e essas etapas seguirão o fluxo de execução do sistema apresentado na Figura 1.



Fig. 1. Metodologia adotada.

Na concepção inicial do trabalho encontramos apenas *datasets* públicos de expressões em LIBRAS, porém mesmo sendo públicos estes *datasets* ora não estavam em formatos amigáveis para o uso em nosso trabalho, ora precisavam de aprovação da instituição detentora dos arquivos e as mesmas não responderam as solicitações. Por isso, para desenvolver e testar o sistema de reconhecimento de LIBRAS, optamos

então, por construir o próprio *dataset* baseado em quadros extraídos de vídeos.

Os vídeos foram gravados no próprio ambiente de desenvolvimento, sendo assim um ambiente com pouco controle. Dada as circunstâncias, decidiu-se escolher o alfabeto como ponto de partida, por ser a base de qualquer linguagem e por ter características suficientes para representar o problema, já que mesmo neste conjunto pequeno de sinais pode-se notar variedade de gestos com as mãos e semelhança entre alguns sinais. É possível destacar por exemplo, a semelhança entre os sinais referentes as letras A e E, ou as letras M e N, como mostra a Figura 2.



Fig. 2. Representações das letras A,E,M e N.

Com isso, as gravações resultaram em oitenta e sete mil *frames* referentes a todas as letras do alfabeto junto com dois sinais de controle, sendo um para significar possíveis espaços entre caracteres e o outro para deletar caracteres. Além destes incluiu-se também imagens com ausência de sinais, ou seja, imagens que representam o fundo.

Durante o desenvolvimento deste trabalho, foram procurados bancos de imagens públicos, que contivessem sequências de imagens (vídeos) relacionadas ao alfabeto de LIBRAS, mas nenhum banco público foi encontrado com essas características, sendo por isso necessária esta etapa.

### A. Sobre o pré-processamento

Como o objetivo final é realizar o mínimo de interferência computacional nas etapas iniciais do processo, foi assegurado apenas que a amostra tentasse representar o máximo possível a realidade. Ou seja, a aquisição das imagens se deu em ambientes com luminosidade diversificada.

Como o interesse está apenas nas configurações de mão, os *frames* limitaram-se apenas à região das mãos. Em seguida foi realizada a retirada de fundo e após a conclusão dessa etapa, foi realizada a extração de características.

### B. Remoção do fundo da imagem

Para a remoção do fundo da imagem, inicialmente foi realizada a conversão do espaço de cores original para outro. No caso de uma transformação para o espaço de cores RGB, a ordem dos canais deve ser especificada explicitamente (RGB ou BGR). Para esta tarefa utilizou-se a biblioteca *OpenCV*. O formato de cor padrão no *OpenCV* é geralmente chamado de RGB, mas na verdade é BGR (os *bytes* são invertidos).

Portanto, o primeiro *byte* em uma imagem colorida padrão (24 bits) será um componente azul de 8 bits, o segundo *byte* será verde e o terceiro *byte* será o componente vermelho. O quarto, quinto e sexto *bytes* seriam então o segundo pixel (azul, depois verde e depois vermelho) e assim por diante.

As faixas convencionais para os valores dos canais R, G e B são:

- 0 a 255 para CV\_8U imagens
- 0 a 65535 para CV\_16U imagens
- 0 a 1 para CV\_32F imagens

No caso de transformações lineares, o intervalo não importa. Porém, no caso de uma transformação não linear, uma imagem RGB de entrada deve ser normalizada no intervalo de valores apropriado para obter os resultados corretos. Se esta função for utilizada em imagens de 8 bits, a conversão terá algumas informações perdidas. Para muitos aplicativos, isso não será perceptível, mas é recomendável usar imagens de 32 bits em aplicativos que precisam de toda a gama de cores ou que convertam uma imagem antes de uma operação e depois convertam novamente.

Então a transformação de RGB para escala de cinza pode ser representada da seguinte forma:

$$Y = 0.299 * R + 0.587 * G + 0.114 * B$$

Após esta etapa é aplicado um filtro gaussiano. No processamento de imagens o filtro gaussiano (também conhecido como suavização gaussiana) é o resultado do desfoque de uma imagem por uma função gaussiana, que é muito utilizado para a redução de ruídos de uma imagem, ou seja, com ele busca-se aprimorar as estruturas de imagem em diferentes escalas.

Matematicamente, este filtro utiliza uma função gaussiana, que expressa a distribuição normal no campo da estatística para calcular a transformação a ser aplicada em cada pixel da imagem de entrada. Em duas dimensões podemos expressar um gaussiano da seguinte forma:

$$G_0(x, y) = A e^{-\frac{(x-\mu_x)^2}{2\sigma_x^2} - \frac{(y-\mu_y)^2}{2\sigma_y^2}} \quad (1)$$

Onde  $x$  é a distância da origem no eixo horizontal,  $y$  é a distância da origem no eixo vertical e  $\sigma$  é o desvio padrão da distribuição gaussiana. Quando aplicada em duas dimensões, essa fórmula produz uma superfície cujos contornos são círculos concêntricos com uma distribuição gaussiana a partir do ponto central. Os valores dessa distribuição são usados para construir uma matriz de convolução aplicada à imagem original.

O novo valor de cada pixel é definido como uma média ponderada da vizinhança desse pixel. O valor do pixel original recebe o peso mais significativo (com o valor gaussiano mais alto) e os pixels vizinhos recebem pesos menores à medida que aumenta a distância do pixel original. Isso resulta em um desfoque que preserva limites e bordas melhor que outros filtros de desfoque mais uniformes, como ilustrado visualmente na Figura 3.



Fig. 3. Filtro gaussiano com *threshold*.



Como este trabalho utiliza a transferência de aprendizado, as configurações para esses passos foram definidas pela arquitetura adotada (*InceptionV3*). Para cada magnitude foram gerados os vetores de características com base no modelo interno de extração de características da arquitetura utilizada.

A figura 6 mostra a camada de extração de características da arquitetura utilizada.

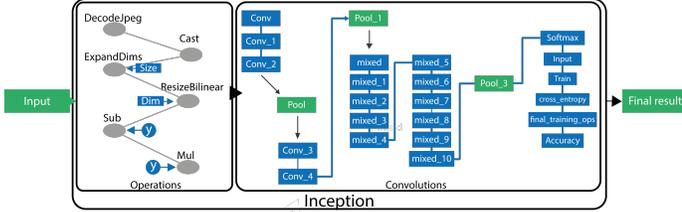


Fig. 6. Gráfico gerado pela arquitetura utilizada.

#### IV. RECONHECIMENTO

A função de ativação *Softmax* é utilizada nessa etapa por ser uma função de classificação que deve ser aplicada na camada de saída da rede. Esta função também é um tipo de função sigmóide, mas é útil nos problemas de classificação. Esta função pode ser definida como:

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad (2)$$

A vantagem do uso da *softmax* é o intervalo de probabilidade de saída com valores entre 0 e 1. Por exemplo, para as saídas como [1.2, 0.9, 0.75] quando aplicada a função *softmax*, obtêm-se [0.42, 0.31, 0.27] e isso essencialmente dá a probabilidade de identificar a qual classe pertence a imagem de entrada. Como o objetivo é gerar as probabilidades para definir a classe de cada entrada, esta foi a função escolhida.

#### V. RESULTADOS DO TREINAMENTO DA REDE NEURAL E INTERFACE FINAL

Como citado anteriormente, o experimento utilizou uma base de imagens do alfabeto em LIBRAS, somando-se a dois sinais de controle (uma para deleção e outro para representar um espaço).

Para a realização dos testes foram selecionadas aleatoriamente 70% da amostra para treino e 30% para teste, de forma semelhante ao que foi feito por Spanhol [10]. Durante esta fase, para que fosse possível definir um critério de parada, observou-se os valores de acurácia e de *cross-entropy*, tanto no conjunto de treino, quanto no de teste. Após atingir o número de 30.000 *steps*, foi possível observar os seguintes resultados de treino e teste, como mostra a Figura 7.

Para definir o melhor resultado deve-se observar os valores de acurácia e *cross-validation* em conjunto. Como pode-se notar, o melhor resultado ocorreu com 25.000 *steps*, como mostra a Tabela I:

A matriz de confusão para este trabalho pode ser verificada na Figura 8. Porém Quando estamos lidando com muitas

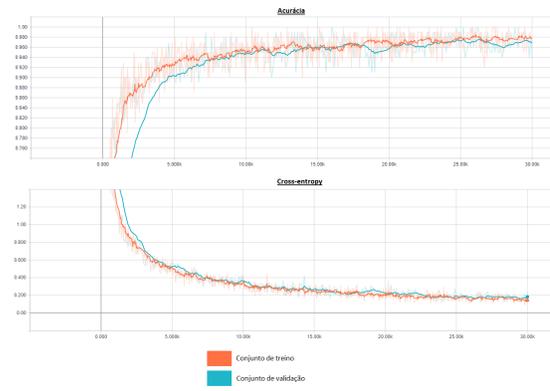


Fig. 7. Gráfico de acurácia e *cross-validation* nos conjuntos de treino e validação.

TABLE I  
QUADRO DE TREINO E TESTE

steps	Acc./treino	Acc./valid.	cross-ent./treino	cross-ent./valid.
10.000	0.89	0.96	0.3853	0.3434
15.000	0.97	0.98	0.2267	0.2801
20.000	0.97	0.96	0.1776	0.2209
25.000	0.99	0.97	0.1446	0.1813
30.000	0.98	0.94	0.1240	0.2347

classes é difícil apenas verificar a matriz de confusão devido a dificuldade de visualização. Por esse motivo podemos verificar outras métricas para avaliar o modelo treinado.

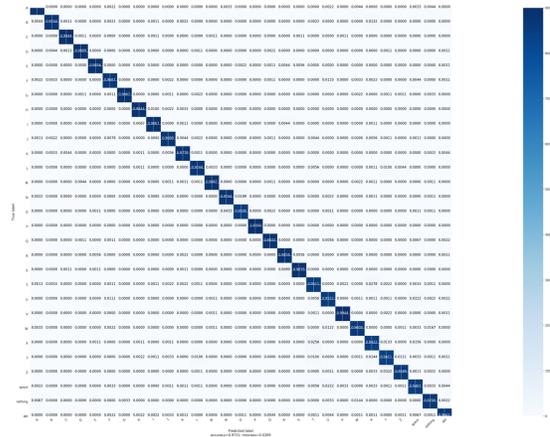


Fig. 8. Matriz de confusão.

As métricas de precisão, *recall*, *f1* score e especificidade para cada classe de nosso modelo são encontrados na Figura 9.

#### A. Interface

Para realizar o experimento de maneira interativa o desenvolvimento da ferramenta foi feito na linguagem *python*, utilizando o ambiente integral de desenvolvimento da empresa *jetbrains* chamada *pycharm*. Essa escolha foi feita por conta

	precision	recall	f1-score	support
A	0.98	0.98	0.98	900
B	0.98	0.97	0.98	900
C	0.99	0.99	0.99	900
D	0.99	0.99	0.99	900
E	0.99	0.99	0.99	900
F	0.97	0.97	0.97	900
G	1.00	0.99	0.99	900
H	0.99	0.98	0.98	900
I	0.98	0.99	0.99	900
J	0.98	0.97	0.97	900
K	0.98	0.98	0.98	900
L	0.98	0.97	0.98	900
M	0.99	0.99	0.99	900
N	0.95	0.97	0.96	900
O	0.98	0.95	0.96	900
P	1.00	1.00	1.00	900
Q	0.99	0.98	0.98	900
R	0.98	0.99	0.98	900
S	0.99	0.99	0.99	900
T	0.94	0.95	0.95	900
U	0.93	0.95	0.94	900
V	0.99	0.99	0.99	900
W	0.97	0.96	0.96	900
X	0.93	0.94	0.93	900
Y	0.93	0.94	0.94	900
Z	0.98	0.96	0.97	900
space	0.93	0.95	0.94	900
nothing	0.96	0.97	0.96	900
del	0.98	0.97	0.97	900
accuracy			0.97	26100
macro avg	0.97	0.97	0.97	26100
weighted avg	0.97	0.97	0.97	26100

Fig. 9. Métricas de avaliação do modelo.

da ampla documentação da linguagem, além de ser linguagem amplamente utilizada por desenvolvedores em problemas de visão computacional e aprendizado de máquina.

Na construção também utilizamos a biblioteca *OpenCV* (Open Source Computer Vision), que consiste em uma biblioteca de domínio público, para desenvolvimento de algoritmos de visão computacional e aprendizado de máquina, licenciada pela licença BSD e amplamente utilizada por empresas e grupos de pesquisa. Esta biblioteca suporta diversas interfaces como C++, C, *Python*, *Java* e *Matlab*, podendo ser utilizado em praticamente todos os sistemas operacionais disponíveis no mercado.

A biblioteca possui módulos de processamento de imagens e vídeos, estrutura de dados, álgebra linear, interface gráfica para o usuário (GUI), controle de mouse e teclado, além de mais de 2.500 algoritmos, muitos dos quais considerados estado da arte, tais como os de segmentação, reconhecimento de faces, aprendizado de máquina, filtragem de imagens, rastreamento de movimento, entre outros.

O sistema captura a imagem com o auxílio de uma *webcam* e ao reconhecer um símbolo por três quadros consecutivos com uma confiança superior a 80%, ele o transcreve em tela, como mostra a Figura 10.

Com base na revisão sistemática realizada anteriormente, reunimos em um quadro comparativo os trabalhos julgados importantes e com objetivos similares quanto à abordagem adotada.

Da Figura 11, verifica-se os trabalhos onde se encontram os melhores resultados em acurácia, que em sua maioria utilizaram algum dispositivo auxiliar. Além do mais, constata-se que dentre as abordagens em LIBRAS, este trabalho atingiu um desempenho superior aos outros métodos mostrados, mesmo com menor poder computacional devido ao uso de redes neurais convolucionais.

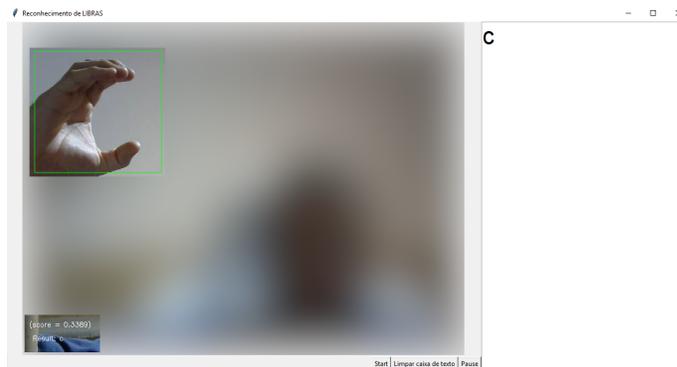


Fig. 10. Interface final construída.

Referência	Método de aquisição	Idioma	Classificação	Acurácia	Interativo
Assaleh [11]	Vídeos em alta resolução	Árabe	Naive bayes	98%	Não
Sandjaja [12]	Vídeos com luvas	Filipino	HMM	85.52%	Não
Highway [13]	Vídeos com roupa e fundo controlados	Australiano	HMM	97%	Não
Starner [14]	Vídeos com luvas	Americano	HMM	99.2%	Sim
Isaacs [15]	Vídeos em estúdio	Americano	ANN	99.9%	Não
Paulraj [16]	Vídeos com roupa e fundo controlados	Kod (Malaysia)	ANN	81.07%	Não
Paulraj [17]	Vídeos com roupa e fundo controlados	Americano	ANN	92.07%	Não
Quan [18]	Vídeos em fundo branco	Chinês	SVM	95.55%	Não
Madeo [19]	Vídeos com roupa e fundo controlados	LIBRAS	Comitê	91.72%	Não
Montero [20]	Vídeos de verbetes	LIBRAS	k-nn	75%	Não
Padovani [21]	Vídeos com roupa e fundo controlados	LIBRAS	HMM	80.1%	Não
Gonçalves [22]	Vídeos com luvas	LIBRAS	Fuzzy	84.3%	Sim
Este trabalho	Frames de vídeos	LIBRAS	CNN	97%	Sim

Fig. 11. Quadro comparativo.

Este desempenho também se deve à técnica de transferência de aprendizado a partir de *Deep learning* treinada para o reconhecimento do conjunto *ImageNet*, pois a profundidade destas redes influencia nos resultados obtidos por classificadores, que receberam como dado de treino os vetores de características gerados a partir da extração feita por essas redes neurais previamente treinadas.

## VI. CONCLUSÃO

O objetivo deste trabalho foi o desenvolvimento de um sistema interativo para reconhecimento de LIBRAS, capaz de, com o auxílio de uma *webcam*, reconhecer sinais expressos e transcrever a palavra correspondente em língua portuguesa, isso sem a necessidade de auxílio de dispositivos eletromecânicos ou qualquer aparato junto ao corpo do usuário. Desta maneira pretende-se facilitar a comunicação entre surdos e/ou deficientes auditivos e pessoas que não conhecem uma língua gestual, proporcionando uma maior inclusão daqueles com o restante da sociedade. Foi realizada uma revisão sistemática, onde foi feita a comparação entre as abordagens adotadas nos trabalhos julgados importantes da área do presente trabalho e a abordagem adotada. Constatou-se que dentre as abordagens em LIBRAS, este trabalho atingiu um desempenho superior aos outros métodos.

O uso de redes neurais convolucionais como também a aplicação de *Deep learning* para a transferência de aprendizado, nos permite concluir que a transferência de conhecimento é uma boa técnica a ser explorada, pois apresentou bons resultados em comparação aos demais estudos. Quanto mais se preservar os dados das imagens originais, como é o caso da rede *InceptionV3* que tem como entrada maiores dimensões de imagem, melhores são os resultados.

No desenvolvimento deste trabalho algumas dificuldades foram encontradas, como a ausência de um banco público de vídeos com execução do alfabeto em LIBRAS, limitando assim o tamanho do *dataset* utilizado no desenvolvimento do trabalho. Além disso, ao focar em trabalhos específicos para a língua portuguesa, praticamente todos se utilizaram da abordagem visual com restrições mecânicas, e/ou não se tratavam de sistemas interativos.

Apesar dos percalços encontrados, o objetivo de criar um sistema interativo com baixo custo computacional foi alcançado, sendo possível através do protótipo criar palavras a partir do alfabeto treinado, sem a necessidade de nenhum aparato auxiliar instalado no corpo do usuário. Acredita-se que todas as etapas do sistema proposto são contribuições para trabalhos futuros da área de reconhecimento de sinais, ou até mesmo trabalhos envolvendo processamento de imagens.

Devido ao tempo, complexidade do assunto em questão e a problemas enfrentados durante o desenvolvimento do trabalho, não era objetivo que o sistema desenvolvido fosse uma solução definitiva para reconhecimento de LIBRAS, e sim uma ferramenta para iniciar um estudo e ser aprimorado em trabalhos futuros.

Como trabalhos futuros, é sugerido se valer do desempenho da transferência de conhecimento feita a partir de bases de dados correlatas, e utilizar as técnicas de *fine-tuning* e *data augmentation* a fim de melhorar os modelos gerados pelas redes profundas.

Além disso, os testes realizados no sistema proposto utilizaram apenas os sinais referentes às letras do alfabeto em LIBRAS. Sendo assim, recomenda-se expandir o conjunto de palavras, assim como a diversificação dos executantes dos sinais. Outra vertente do trabalho seria tornar o sistema desenvolvido em uma aplicação mobile, possibilitando o uso do mesmo em qualquer dispositivo móvel.

Por fim, este trabalho focou nos parâmetros de configuração de mão, movimento e orientação para o reconhecimento dos sinais. Uma outra extensão, pouco tratada nos trabalhos relacionados e tampouco se encontra na literatura, seria as expressões não manuais como parâmetro para a composição dos sinais.

## REFERENCES

- [1] World Health Organization, "Deafness and Hearing Loss", pp. 1–4, 2018.
- [2] da Silva Borges, André and Aparecida de Araujo Xavier, Eliane and Bercovich, Alicia, "Censo demográfico 2010", vol. 27, 2011.
- [3] Stokoe, William C. and Marschark, Marc, "Sign language structure: An outline of the visual communication systems of the american deaf", Journal of Deaf Studies and Deaf Education, pp. 3–37, vol. 10, 2005.
- [4] Digiampietri, Luciano A. and Teodoro, Beatriz and Santiago, Caio R.N. and Oliveira, Guilherme A. and Araujo, Jonatas C., "Um Sistema de Informação Extensível Para o Reconhecimento Automático de LIBRAS", VIII Simpósio Brasileiro de Sistemas de Informação (SBSI 2012), pp. 456–467, 2012.
- [5] Bianco, Simone and Cadene, Remi and Celona, Luigi and Napoletano, Paolo, "Benchmark analysis of representative deep neural network architectures", IEEE Access, pp. 64277–64277, vol. 6, 2018.
- [6] Christian Szegedy et al., "Going Deeper with Convolutions", University of Michigan, pp. 163–182, 2019.
- [7] Thomas Serre, Lior Wolf, Stanley Bileschi, Maximilian Riesenhuber, and Tomaso Poggio, "Robust Object Recognition with Cortex-Like Mechanisms", IEEE transactions on pattern analysis and machine intelligence, pp. 102–104, vol. 38, 2016.
- [8] Lin, Min and Chen, Qiang and Yan, Shuicheng, "Network in network", 2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings, pp. 1–10, 2014.
- [9] Szegedy, Christian and Vanhoucke, Vincent and Ioffe, Sergey and Shlens, Jon and Wojna, Zbigniew, "Rethinking the Inception Architecture for Computer Vision", Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2818–2826, vol. 2016, 2016.
- [10] Spanhol, Fabio A. and Oliveira, Luiz S. and Petitjean, Caroline and Heutte, Laurent, "A Dataset for Breast Cancer Histopathological Image Classification", IEEE Transactions on Biomedical Engineering, pp. 1455–1462, vol. 63, 2016.
- [11] Assaleh, Khaled and Shanableh, Tamer and Fanaswala, Mustafa and Bajaj, Harish and Amin, Farnaz, "Vision-based system for continuous arabic sign language recognition in user dependent mode", pp. 19–27, vol. 2010, 2008.
- [12] Sandjaja, Iwan Njoto and Marcos, Nelson, "Sign language number recognition", NCM 2009 - 5th International Joint Conference on INC, IMS, and IDC, pp. 1503–1508, 2009.
- [13] Goh, Paul and Holden, Eun-Jung, "DYNAMIC FINGERSPELLING RECOGNITION USING GEOMETRIC AND MOTION FEATURES School of Computer Science & Software Engineering The University of Western Australia", Image Processing, 2006 IEEE International Conference on, pp. 2741–2744, 2006.
- [14] Starner, Thad and Pentland, Alex, "Real-Time American Sign Language Recognition from Video Using Hidden Markov Models", pp. 227–243, 1997.
- [15] Isaacs, Jason and Foo, Simon, "Optimized wavelet hand pose estimation for American sign language recognition", Proceedings of the 2004 Congress on Evolutionary Computation, CEC2004, pp. 797–802, vol. 1, 2004.
- [16] Paulraj, M. P. and Yaacob, Sazali and Desa, Hazry and Majid, Wan Mohd Ridzuan Wan Ab, "Gesture recognition system for Kod Tangan Bahasa Melayu (KTBM) using neural network", Proceedings of 2009 5th International Colloquium on Signal Processing and Its Applications, CSPA 2009, pp. 19–22, 2009.
- [17] Paulraj, M. P. and Yaacob, Sazali and Desa, Hazry and Hema, C. R. and Wan Ab Majid, Wan Mohd Ridzuan, "Extraction of head and hand gesture features for recognition of sign language", 2008 International Conference on Electronic Design, ICED 2008, 2008.
- [18] Quan, Yang, "Chinese sign language recognition based on video sequence appearance modeling", Proceedings of the 2010 5th IEEE Conference on Industrial Electronics and Applications, ICIEA 2010, pp. 1537–1542, 2010.
- [19] Madeo, Renata C. B. and Peres, Sarajane M. and Bísvaro, Helton H. and Dias, Daniel B. and Boscaroli, Clodis, "A Committee Machine Implementing the Pattern Recognition Module for Fingerspelling Applications", pp. 954–958, 2010.
- [20] Henrique, Carlos and Monteiro, A and Felipe, Luiz and Pecoraro, Inácio and Lacerda, Angélica Takamine and Corbo, Anna Regina and Matos Araujo, Gabriel, "Um sistema de baixo custo para reconhecimento de gestos em LIBRAS utilizando visão computacional"
- [21] Kleber Padovani de Souza, Jéssica Barbosa Dias, Hemerson Pistori, "Reconhecimento Automático de Gestos da Língua Brasileira de Sinais utilizando Visão Computacional", Lecture Notes in Mathematics, pp. 59–95, vol. 2255, 2019.
- [22] Gonc, Wagner M and Peres, Sarajane M and B, Rua Arlindo and S, Av Trabalhador, "Funções de Similaridade em CBIR no Contexto de Teste de Desempenho de Funções de Imagens de Gestos da LIBRAS Software : Estudo de Caso em Segmentação", pp. 6, 2018.