



IDENTIFICAÇÃO DOS MOVIMENTOS REALIZADOS PELA MÃO DURANTE A FALA EM LIBRAS PARA AUXILIAR NA COMUNICAÇÃO DOS SURDOS

Susan Schultz Thiel¹

Cleber Lourenço Izidoro²

Resumo: A falta de compreensão da linguagem de sinais por meio de grande parte da população ouvinte obstrui a comunicação entre este grupo com a população surda. Por meios convencionais este diálogo estaria limitado à linguagem escrita ou a presença de intérpretes, soluções estas que podem se tornar inconvenientes, onerosas e retiram a privacidade e independência em determinadas situações. Novas tecnologias possibilitam que diálogos em fala sejam reconhecidos e transformados em texto ou animações, contudo a identificação do que é expressado na linguagem de sinais ainda é uma área em desenvolvimento. Buscou-se com este estudo a utilização do conceito de robótica assistiva para a confecção de um protótipo capaz de registrar a orientação espacial da mão e a configuração de mão exercida pela compressão dos dedos a fim de que estas informações fossem utilizadas no treinamento de uma rede neural com a finalidade de reconhecer expressões ditas na Língua Brasileira de Sinais (LIBRAS). Deste modo utilizou-se uma estrutura mecânica fabricada em impressão 3D na qual a coleta dos valores referentes à orientação espacial é realizada pelo sensor MPU-6050, integrado de um acelerômetro e um giroscópio tipo MEMS. Já a compressão do dedo indicador é indicada através de um potenciômetro associado ao mecanismo. Os dados recolhidos foram utilizados para treinamento e validação de uma rede neural artificial com uma camada oculta de arquitetura LSTM (*Long Short-Term Memory*), responsável por auxiliar o modelo computacional a reconhecer qual informação deverá ser armazenada ou descartada à memória de longo prazo. O presente trabalho obteve uma precisão aproximada de 62,82% na detecção dos movimentos analisados, os quais eram reproduzidos em forma de áudio após a identificação. Desta forma foi possível observar a capacidade do modelo confeccionado de reconhecer alguns padrões de movimentação durante a fala em LIBRAS.

Palavras-Chave: LIBRAS. Análise de Movimentos. *Wereble*. Redes Neurais. LSTM.

1 INTRODUÇÃO

A comunicação é fundamental para a convivência em sociedade, contudo para a parcela surda e/ou muda da população existe uma barreira devido à falta de compreensão da linguagem de sinais por meio da população ouvinte. Diversos fatores

¹ Graduando em Engenharia Mecatrônica. Ano 2021/1 E-mail: susan@thiel.eng.br

² Professor do Centro Universitário UniSATC. E-mail: cleber.izidoro@satc.edu.br

podem levar um indivíduo a ser surdo, dentre anomalias congênitas, tratamento inadequado de doenças como meningite, tuberculose e de infecção no ouvido ou devido a ruídos e sons elevados (COLLADO, 2016). Atualmente, segundo dados do Instituto Locomotiva e da Semana da Acessibilidade Surda (GANDRA, 2019), aproximadamente 10,7 milhões de pessoas possuem deficiência auditiva no território brasileiro.

As linguagens de sinais, assim como a linguagem oral, não são universais. Desta forma a Língua Brasileira de Sinais (LIBRAS) possui conjuntos de palavras e alfabeto diferentes, por exemplo, da língua de sinais americana (*American Sign Language – ASL*) e da Língua Alemã de Sinais (*Die Deutsche Gebärdensprachen – DGS*). A Língua Brasileira de Sinais é descrita por Silva e Jesus (2017), como uma linguagem psíquica visual, na qual são utilizados gestos, movimentos e expressões faciais. Dentro dos parâmetros da LIBRAS, há 111 configurações de mão, às quais somadas ao movimento, localização e orientação da mão, compõem as frases.

Enquanto há ferramentas de reconhecimento de voz que possibilitam transformar a fala em texto ou animações 3D (*Hand Talk*, 2020), a comunicação no sentido de surdos para ouvintes se limita à linguagem escrita ou à presença de intérpretes. Entretanto, a linguagem escrita é considerada inconveniente, especialmente em situação de deslocamento e quando o interlocutor está distante. Já a assistência de intérpretes costuma demandar alto custo e ocasiona a falta de privacidade e independência (OZ; LEU, 2011).

Uma das formas de auxiliar na comunicação de surdos poderá ser realizada através da Robótica Assistiva (RA). Este conceito é referente a aplicações da robótica que contribuem na redução de problemas e que desenvolvam habilidades funcionais a pessoas com deficiência. Deste modo, pesquisas referentes a este tema visam fornecer adaptações como: correção da postura, mobilidade, equipamentos de assistência à visão e ampliação de habilidades de comunicação (EUGENIO, 2016). Neste contexto, a identificação dos movimentos realizados pelas mãos durante a comunicação em LIBRAS possibilitaria que um dispositivo reproduzisse na língua falada a frase comunicada.

Com esse intuito, autores como Park E, Sehdev N. e Frogoso R. (2016) já desenvolveram aplicações baseadas em reconhecimento de imagens, pouco aplicáveis em situações cotidianas, por consequência da necessidade de obtenção das imagens. Já trabalhos, como o de Lubbadeh M. e Shashtari A. (2017), utilizam

sensores de posicionamento conhecidos como acelerômetros, juntamente com sensores do tipo flexível, os quais alteram sua característica de resistência elétrica, por meio da aplicação de uma força decorrente da flexão, entretanto estes são de difícil aquisição.

Posto isso, a proposta deste trabalho é a elaboração de um dispositivo capaz de reconhecer a movimentação e orientação da mão através da apuração dos dados obtidos por um acelerômetro triaxial. Em paralelo, um mecanismo produzido em impressora 3D, associado à eletrônica será responsável por coletar informações relacionadas às configurações de mão executadas. Desta forma, ao associar-se ambas as informações, será realizada a interpretação da mensagem transmitida na Língua Brasileira de Sinais, possibilitando uma nova alternativa para o diálogo entre uma pessoa surda e alguém que não compreenda LIBRAS.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo serão apresentados estudos e publicações científicas, referentes ao tema de identificação dos movimentos, a partir de um sensor de posicionamento, assim como a análise necessária para o reconhecimento de frases expressas em linguagens de sinais.

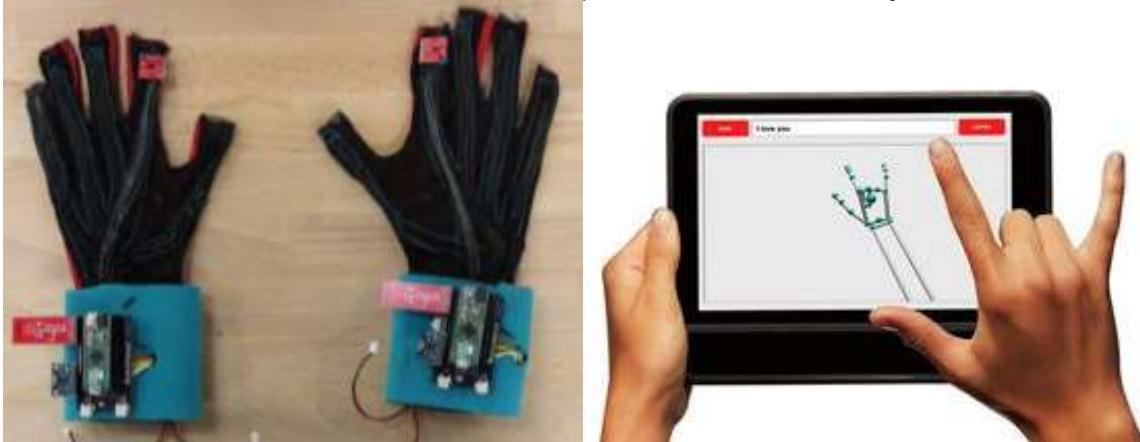
2.1 DISPOSITIVOS PARA A TRADUÇÃO DE LINGUAGENS DE SINAIS

Conforme apresentado por Oz e Leu (2011), pesquisas relacionadas à identificação de linguagem de sinais podem ser majoritariamente divididas em três classes principais: a) baseado em reconhecimento de imagens, b) luvas compostas por sensores de movimentos, e c) uma combinação dos métodos anteriores.

O projeto *SignAloud*, desenvolvido por Thomas Pryot e Navid Azodi, realiza a tradução da Língua Americana de Sinais (ASL) para áudio ou texto, conforme a segunda classe. O dispositivo é composto por um par de luvas com sensores, demonstradas na Fig. 1(a), que fazem o reconhecimento dos gestos e enviam as informações, via Bluetooth, para o computador que realiza a comparação dos sinais obtidos com os movimentos salvos através de aprendizado de máquina (*Machine Learning*). Após o reconhecimento da frase esta é pronunciada através de uma voz computadorizada (LUBBADEH; SHASHTARI, 2017).

Por outro lado, o produto *MotionSavvy* utiliza um dispositivo eletrônico *Leap Motion*, composto por câmeras e infravermelho para realizar o reconhecimento das mãos do usuário e, assim, gerar em tempo real um modelo 3D, como indica a Fig. 1(b). Logo após, a frase é identificada e, então, reproduzida em áudio ou texto.

Figura 1: (a) Protótipo do *SignAloud* e (b) Produto *MotionSavvy*



Fonte: (a) O'Hare (2016) e (b) Bancaleiro (2014)

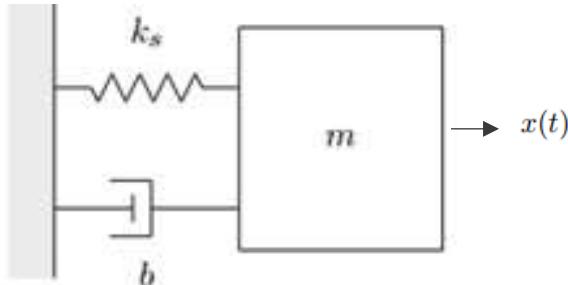
2.2 SENSOR DE ACELERAÇÃO

A captação de variações físicas para a geração de impulsos ou sinais elétricos é realizada com a utilização de elementos físicos conhecidos como sensores. A escolha desses equipamentos para um projeto de instrumentação decorre do conhecimento de seu funcionamento e das especificações do projeto.

2.2.1 Princípio de funcionamento

O acelerômetro é um dispositivo eletromecânico que realiza a mensuração da aceleração, ou seja, a taxa de variação da velocidade. A aceleração é resultante da aplicação de uma força sobre um corpo, desta forma o elemento central deste equipamento é uma massa m suspensa por uma estrutura elástica constituída por amortecedor e mola, como demonstra a Fig. 2.

Figura 2: Sistema massa-mola-amortecedor básico



Fonte: Florêncio (2015)

Dessa forma, a força a que é submetida a estrutura é repassada à massa “m” e um sistema de detecção gera uma diferença de potencial em seus terminais proporcional à força (FLORÊNCIO, 2015). O mais frequente é que sejam utilizados capacitores móveis ou cristais piezoelétricos, os quais criam uma corrente elétrica que atravessa o peso e pode ser medida para determinar a aceleração. Contudo, há outros que utilizam como base modelos físicos como os acelerômetros piezoresistivos, térmicos, servo e ópticos (LU; WANG; LIU, 2018).

As vantagens da aplicação de acelerômetros capacitivos se dão pelo baixo consumo de energia, baixo custo de fabricação e baixa sensibilidade a mudanças na temperatura e conveniente inclusão em circuitos integrados. Contudo é recorrente a necessidade do uso de métodos de filtragem para obtenção de sinais mais limpos. Já acelerômetros piezoelétricos apresentam baixa emissão de resíduo, ampla faixa dinâmica e de frequência, contudo não conseguem medir a aceleração estática e quase estática (LU; WANG; LIU, 2018). A aceleração pode ser classificada dentre estática e dinâmica, referindo-se à força da gravidade e à originária de uma movimentação ou vibração respectivamente (TEVES A. C., 2013).

São disponibilizados no mercado acelerômetros capazes de mensurar a aceleração em um, dois ou três eixos ortogonais, simultaneamente, e são usualmente utilizados para três tipos de atuação: medida inercial de velocidade e posição, sensor de inclinação, rotação ou orientação entre duas ou três dimensões e como sensor de vibração ou impacto. Segundo Teves (2013), as principais especificações que devem ser examinadas durante a escolha de um acelerômetro são:

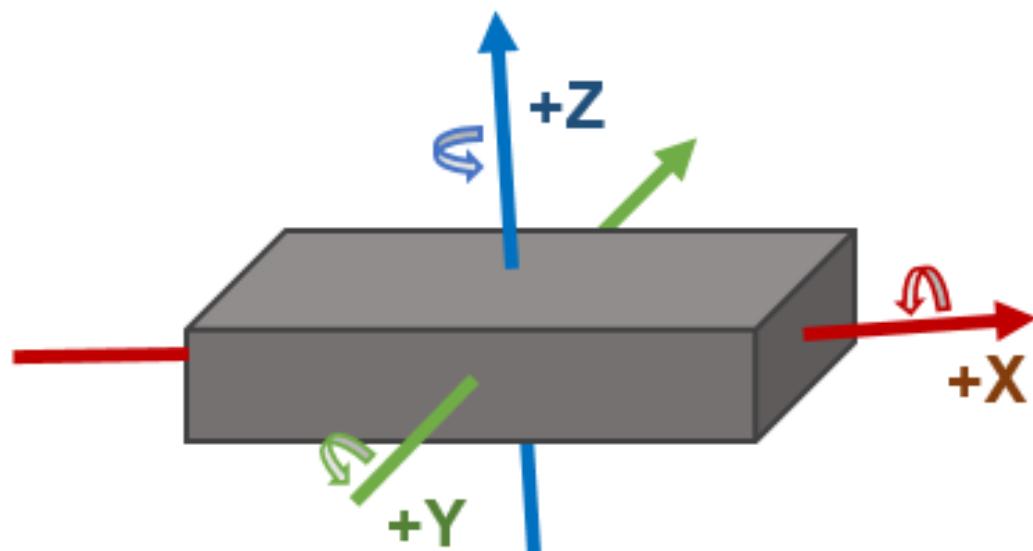
- Sensibilidade;
- Fundo de escala (*Full-scale range*);

- Direção (número de eixos);
- Largura de banda;
- Estabilidade;
- Resolução;
- Alcance dinâmico (*Dynamic Range*);
- Confiabilidade;
- Custo.

No que se refere à direção, Júnior (2018) demonstra que o acelerômetro de uma direção, também classificado como de eixo único, costumava ser o modelo mais aplicado. Contudo com a redução do custo dos acelerômetros de três eixos, exemplificados pela Fig. 3, resultou um maior número de aplicações do mesmo por permitir, em algumas aplicações, diminuir em até um terço o número de sensores na instrumentação.

Com relação à largura de banda necessária é conhecido que os movimentos corporais realizados por seres humanos não são muito superiores à faixa de 10-12 Hz. Dessa forma, conforme o teorema de amostragem de Nyquist-Shannon, uma largura de banda de 24 Hz já seria adequada para este tipo de aplicação, porém usualmente utiliza-se uma frequência de 40-60Hz por resguardo (JÚNIOR, 2018).

Figura 3: Representação dos eixos de um acelerômetro triaxial



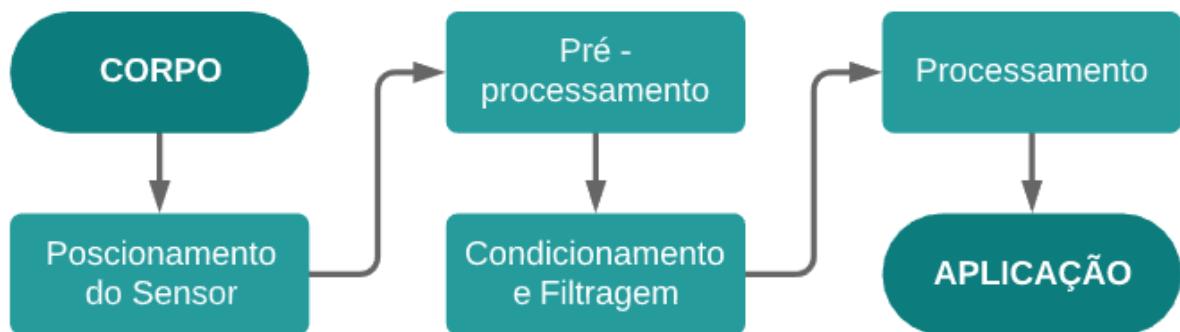
Fonte: Da autora (2021)

2.3 PROCESSAMENTO DOS SINAIS

Como descreve Silva (2013), há seis camadas que podem estar presentes em um sistema acoplado ao corpo, e cada uma representa uma atribuição no processo de reconhecimento de movimentos. A representação destas camadas pode ser observada na Flux. 1. É importante ressaltar que todas as etapas deste sistema não necessitam estar em apenas uma unidade de *hardware*. Dessa forma, os seguimentos presentes em um sistema que realiza o processamento em tempo real dos dados coletados podem ser classificados entre:

- Sistema transmissor de dados: é responsável pela aquisição e transmissão das amostras para um computador local ou remoto, onde será realizado o processamento. Possui a vantagem de ser um sistema com peso, memória e consumo de energia reduzidos, assim como necessitar de um processador mais simples.
- Sistema processador de dados: Desempenha a aquisição, o pré-processamento e a classificação dos movimentos através de um processador de maior poder computacional que costuma demandar mais energia.

Fluxograma 1: Representação em camadas de um sistema acoplado ao corpo



Fonte: Da autora (2021)

2.3.1 Princípio de funcionamento

Dentre as informações adquiridas pelos acelerômetros é perceptível a presença de ruídos que dificultam a medição e a análise dos dados (FRIZON; PECCIN; JINBO, 2015). Devido a isto, a fusão sensorial é utilizada para que, através de um modelo matemático, seja gerado um algoritmo que combina dados, a fim de

fornecer informações mais precisas. Como apontam (FRANCZAK, 2015) os sinais de resposta provenientes do acelerômetro e do giroscópio, possuem interpretações semelhantes e podem ser fundidos matematicamente na finalidade de diminuir erros.

Para a realização da fusão sensorial há dois métodos principais, o filtro de Kalman e o filtro complementar. O filtro de Kalman é utilizado para a estimação de velocidade, posição e orientação através de um conjunto de equações matemáticas que minimizam a matriz de covariância do erro (FRIZON; PECCIN; JINBO, 2015). Em contrapartida, o filtro complementar funciona de forma análoga à junção de um filtro passa-baixa e um filtro passa alta. Dessa forma, a implementação de um filtro complementar é mais fácil e gasta menos ciclos de processamento (FRANCZAK, 2015).

2.3.2 Dispositivos SoC

Sistemas embarcados do tipo SoC (*System on a Chip* ou Sistema em um chip) são recorrentes em aplicações que demandam tamanho reduzido, alta performance e baixa dissipação de energia. Isto se dá, pois, os modelos modernos são provenientes de uma compacta combinação de microprocessadores, DSPs (*Digital Signal Processor*), processadores gráficos, entradas e saídas (IO) de alta velocidade, memória, dentre outros periféricos (GUERTIN, 2018).

Esses componentes possuem sinal misto, digital e analógico, a fim de utilizar as melhores particularidades de cada um. O processamento analógico é mais rápido quando comparado aos atrasos no nível de porta dos sistemas digitais. Os circuitos digitais também possuem diferença entre níveis altos e baixos quando utilizada baixa tensão. Outra particularidade desses componentes é que, dada a complexidade do seu design, os mesmos demandam esforços significantes para seu projeto. Dessa forma, estes componentes são encontrados em aplicações mobile, câmeras fotográficas, *tablets*, computadores, televisões digitais dentre outros (KAMAL, 2020).

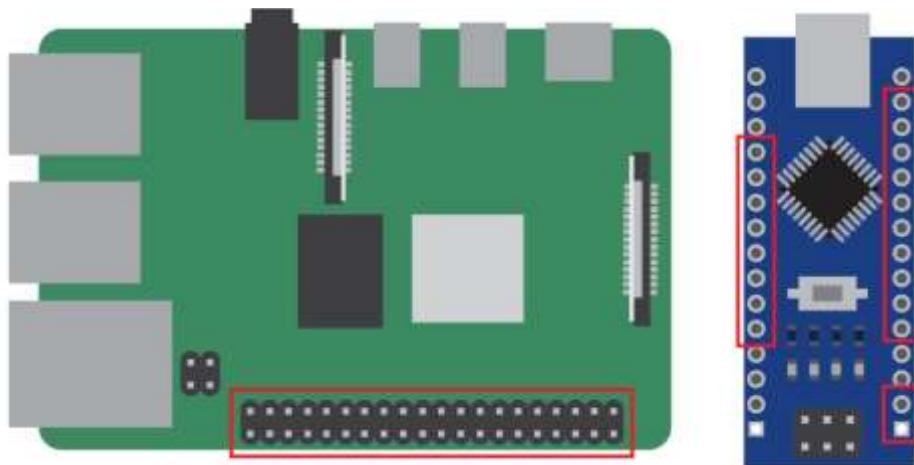
2.3.3 Placas de Prototipagem rápida e *Single-Board Computer*

A utilização de placas de prototipagem em projetos possibilita que os valores de entrada sejam analisados durante a etapa de programação. Ao mesmo

tempo, sistemas de placa única (SBC – *Single-Board Computer*) com microcontroladores ou microprocessadores proporcionam um baixo custo e peso, assim como flexibilidade em suas aplicações (AMBROŽ, 2017).

Como exemplo podem ser citadas, respectivamente, as placas Arduino e Raspberry, que costumam ser utilizadas para o desenvolvimento de sistemas IoT (*Internet of Things*), de forma que o software é inicialmente elaborado utilizando uma IDE (*Integrated Development Environment*) ou IDLE (*Integrated Development and Learning Environment*) e repassado para a memória da placa, onde é testado utilizando o simulador e ferramentas de análise para o protótipo final (KAMAL, 2020). Os pinos GPIO (*General Purpose Input-Output*), existentes nas placas de prototipagem, possibilitam a conexão com circuitos externos. A Fig. 4 ilustra os pinos presentes nestas placas.

Figura 4: Exemplos de pinos GPIO nas placas de prototipagem rápida



Fonte: Da autora (2021)

2.4 PROGRAMAÇÃO

A programação de placas Arduino dá-se através da IDE Arduino Software utilizada em um computador e a sua linguagem é baseada em C/C++. Dada a simplicidade e robustez da placa, a mesma é amplamente utilizada para iniciar os conhecimentos em eletrônica e programação. A linguagem de programação Python é conhecida por ser clara, objetiva e também poderosa (MENEZES, 2014). Ela pode ser empregada em diversas áreas de estudo dentro da Ciência de Dados (Data Science) incluindo o campo da inteligência artificial e suas ramificações: *machine learning*,

redes neurais e *deep learning*. Para a sua utilização é necessário possuir um software interpretador gratuito disponibilizado para os sistemas operacionais Microsoft Windows, Linux e Mac OS X. A programação em Python também pode ser utilizada em outros dispositivos, como por exemplo, a placa Raspberry Pi, o que possibilita a implementação em dispositivos mais portáteis.

2.4.1 Machine Learning

A análise e transformação dos dados em ações inteligentes através do desenvolvimento de algoritmos de computador é conhecida como *Machine Learning*, ou em português, Aprendizado de Máquina. O processo de aprendizagem, tanto para humanos como para máquinas, pode ser subdividido em: recebimento dos dados, abstração e generalização. Dessa forma, no primeiro momento, novas informações são coletadas e analisadas. Estas são abstraídas e assimiladas, a fim de gerar um modelo que torne explícito os padrões estruturados entre os dados. Por fim, na generalização já há um alto nível de compreensão dos dados e como aplicá-los a situações desconhecidas (LANTZ, 2013).

Os dois principais modelos de aprendizado para *machine learning* se distinguem pela existência de um conhecimento prévio, ou não, das respostas corretas (GRUS, 2016). Modelos supervisionados lidam com conjuntos de pares X e Y, que indicam a fonte dos dados e os resultados desejados, respectivamente. Este modelo de aprendizagem é adequado para tarefas que contenham um padrão e um resultado a ser alcançado, como reconhecimento de fala ou classificação de imagens.

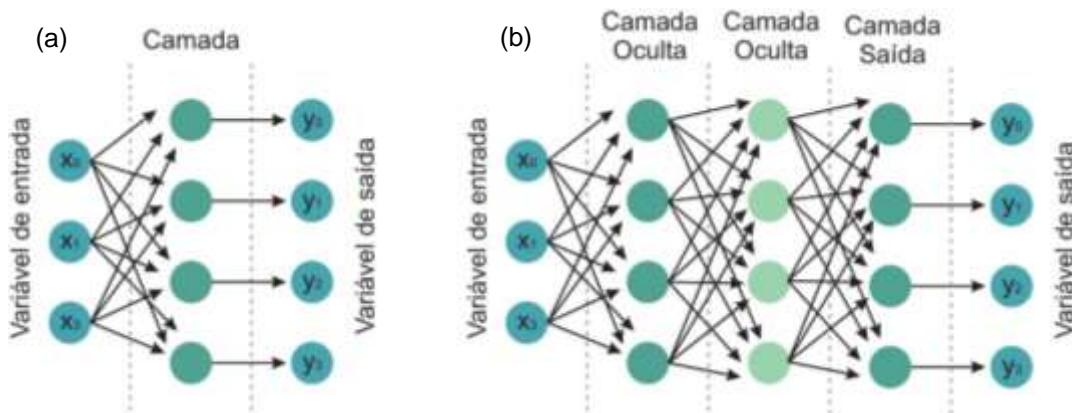
Por outro lado, em modelos de aprendizagem não supervisionada há apenas o conjunto de dados X, de forma que o algoritmo busque extrair as informações relevantes e realizar inferências. À vista disso o aprendizado não supervisionado é aplicado em tarefas de agrupamento, compactação de dados e modelagem estatística (SOARES; SOUZA, 2016).

2.4.2 Redes Neurais Artificiais

As redes neurais artificiais são descritas como um modelo preditivo inspirado pelo funcionamento do cérebro humano, na qual em uma interligada coleção de neurônios a informação apenas é uma transmitida caso cada neurônio tenha sua

condição de ativação alcançada (GRUS, 2016). Conforme a disposição dos neurônios é possível classificar a rede neural entre monocamada e múltiplas camadas, conforme indica a Fig. 5. Este último grupo se destaca na aprendizagem a partir de dados não lineares e aproximação de funções que representem os dados reais (SOARES; SOUZA, 2016).

Figura 5: Diferenciação entre rede neural monocamada (a) e rede neural de múltiplas camadas (b)



Fonte: Adaptado de Soares e Souza (2016)

Na resolução de problemas referentes a séries temporais, como reconhecimento de fala e processamento de linguagem natural, são frequentemente aplicadas Redes Neurais Recorrentes (*Recurrent Neural Networks* - RNNs) as quais exploram as conexões temporais entre os neurônios (NAFEA et al., 2021).

Uma variação denominada de Rede neural de Memória de Longo-Curto Prazo (*Long Short-Term Memory* - LSTM) possui como diferencial que a memória para armazenamento, leitura e gravação são controlados por portas lógicas especiais. Com isto a arquitetura apresenta performance superior, pois consegue definir se a memória existente deve ser mantida ou se novas informações devem ser adicionadas a ela (NAFEA et al., 2021; CHEN et al., 2017).

2.5 MANUFATURA ADTIVA COM IMPRESSÃO 3D

A aplicação dos sensores vestíveis, anteriormente citados, é composta, preferencialmente, por dispositivos pequenos e customizáveis, mas que, ao mesmo tempo, não sejam afetadas pelas conduções do ambiente e a atividades humanas (SU

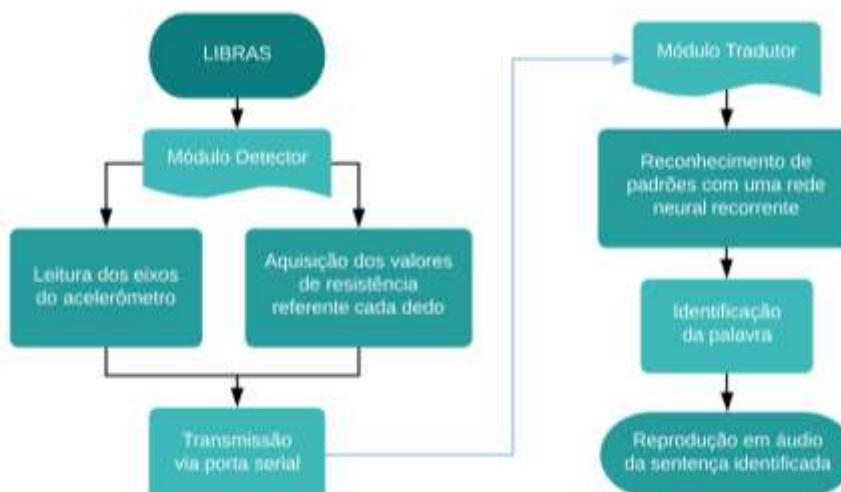
et al., 2017). Dessa forma é perceptível que, para a geração de uma ferramenta para monitoramento e reconhecimento dos movimentos, é necessário que a mesma possa ser ajustável e/ou produzida sob medida para o usuário.

O conceito de manufatura aditiva por impressão 3D permite a confecção de modelos detalhados com apenas uma fração de tempo e custos demandados pelos métodos convencionais de produção (COLELLA; CATARINUCCI, 2018). Em função de se tratar de uma tecnologia bastante flexível, alguns dos exemplos de aplicações são a confecção de sensores vestíveis para aplicações inteligentes de saúde (SU et al., 2017; YOON et. al., 2017), carcaça para sistemas de monitoramento da saúde das articulações (TEAGUE et. al., 2020) e confecção de próteses de braço de baixo custo (CANIZARES; PAZOS; BENITEZ, 2017).

3 PROCEDIMENTOS EXPERIMENTAIS

Com base nos fundamentos apresentados na revisão bibliográfica o protótipo de luva tradutora de línguas de sinais composta por sensores de movimento foi dividido entre dois módulos: detector e tradutor, conforme é indicado pela Fig. 6. O módulo detector é responsável por adquirir os dados referentes a gestos de mão e compressão dos dedos. Já o módulo tradutor recebe as informações provenientes do primeiro módulo e realiza a identificação dos gestos e tradução.

Figura 6: Funcionamento do projeto



Fonte: Autores (2021)

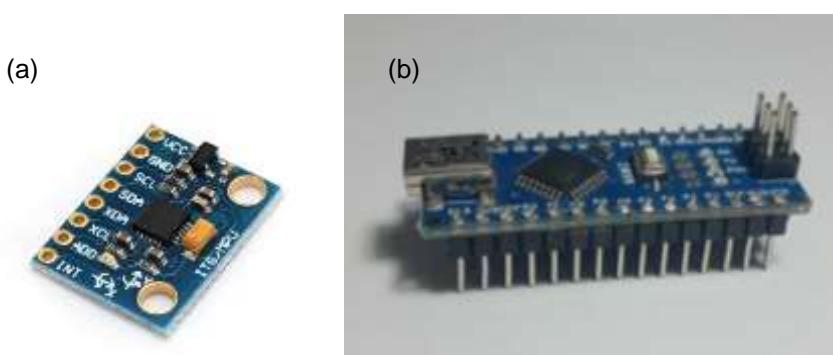
A separação entre os módulos, é dada principalmente pelo processamento deste protótipo ser realizado em um microcomputador. Contudo, o treinamento e/ou aplicação da rede neural já treinada poderia ser realizado em dispositivos mais compactos e, dessa forma, esta divisão resultaria em maior comodidade, visto que o equipamento não seria tão pesado para se tornar portátil.

3.1 CONFECÇÃO DO MÓDULO DETECTOR

A estrutura mecânica utilizada para a coleta dos dados referentes à flexão dos dedos foi baseada na amostra de arquivo SQL para impressão, disponibilizada pela empresa Youbionic (2019). O sensor MPU-6050 foi escolhido para realizar a coleta dos valores para a orientação espacial do protótipo, ilustrado pela Fig. 7(a), visto que este chip possui um acelerômetro e um giroscópio tipo MEMS, ambos com três eixos; dessa forma, são disponibilizados seis graus de liberdade para a análise.

O comando do módulo detector é realizado pelo Arduino Nano, que possui o diferencial de seu tamanho reduzido, fator que permite maior flexibilidade em projetos com esta demanda. Possui o microcontrolador ATmega328, 14 portas digitais (das quais, seis podem ser usadas como PWM), memória flash de 32KB e velocidade de clock de 16MHz.

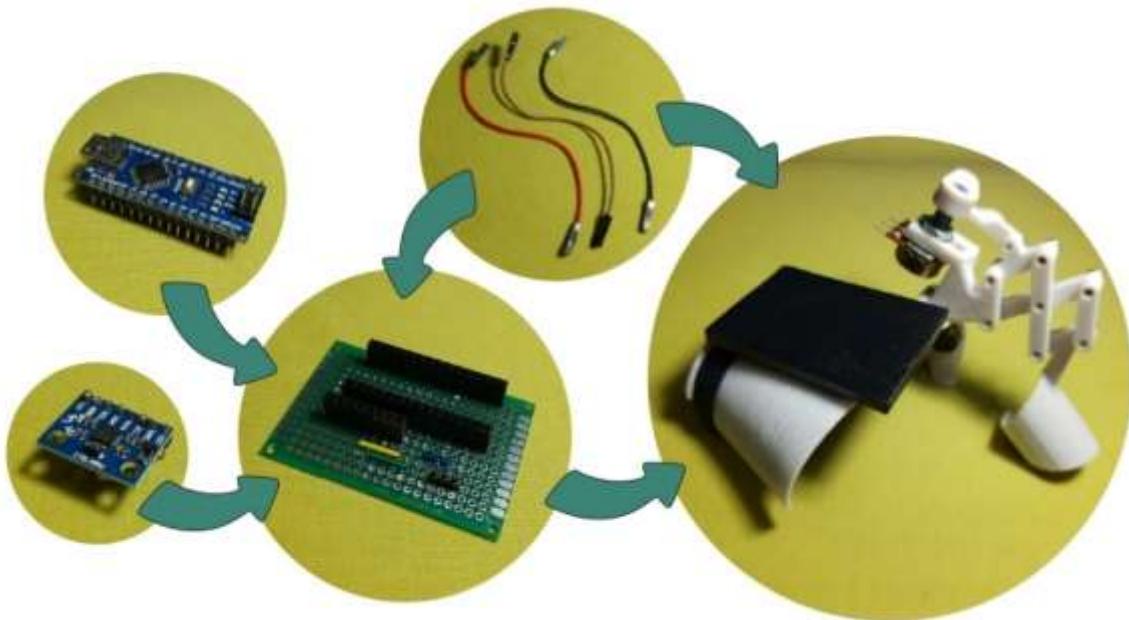
Figura 7: (a) Arduino Nano (b) sensor MPU-6050



Fonte: (a) Arduino (2020) (b) Da autora (2021)

Em uma placa perfurada foram soldadas as trilhas do circuito e os conectores para acoplar o Arduino Nano, o sensor MPU-6050 e os jumpers de ligação com o potenciômetro fixado na estrutura 3D. O esquema da montagem pode ser visualizado pela Fig. 8.

Figura 8: Montagem do protótipo



Fonte: Da autora (2021)

O protótipo verifica os ângulos formados pela compressão do dedo indicador através da resistência resultante no potenciômetro associado ao mecanismo fabricado na impressora 3D, exemplificado pela Fig. 9. Associado a esta estrutura foi acoplado o acelerômetro tri axial que é responsável pela coleta dos valores provenientes da movimentação exercida em cada dos seus eixos. Os dados citados anteriormente são reconhecidos pelo Arduino Nano e enviados ao computador via porta serial.

Figura 9: Protótipo para identificação dos gestos realizados pela mão



Fonte: Da autora (2021)

3.2 COLETA DE DADOS

Com o auxílio da biblioteca i2cdevlib para Arduino, o Processador de Movimento Digital presente no MPU-6050 realiza a combinação dos dados do giroscópio e do acelerômetro para o tratamento dos mesmos. A biblioteca disponibilizava cinco variações entre as leituras das informações fornecidas pelo sensor MPU-6050: o valor puro dos quartenions, os ângulos de Euler calculados a partir do valor quartenion, os componentes de arfagem, guinada e rolagem (*Pitch*, *Yaw* e *Roll*) resultados do valor quartenion e a gravidade calculada, os componentes de aceleração com a gravidade removida e os componentes de aceleração com o desconto da gravidade e a referência da orientação inicial. Dessa forma, optou-se em registrar todas as possíveis variáveis a fim de realizar posterior seleção.

O registro dos valores utilizados, para preencher os bancos de dados para treinamento e teste da rede neural, foi feito através do ambiente de desenvolvimento integrado *PyCharm* via porta serial. O comando programado inicia o registro dos valores de aceleração e ângulo da mão, assim como a flexão do dedo indicador através do valor de resistência.

A janela de tempo escolhida para registro dos movimentos foi de 600 passos, o que corresponde a um intervalo de tempo aproximado de 1200 milissegundos, visto que seria o suficiente para registrar os movimentos referentes às palavras escolhidas para a análise. Por fim, estas informações foram repassadas ao banco de dados no formato CSV (*Comma-Separated-Values*), conforme Tab. 1, que demonstra uma amostra dos valores pertinentes à palavra “olá”.

Tabela 1: Amostra dos valores tocantes à palavra “olá”.

Registro	Gesto	Passo	Potenc.	Euler_x	Euler_y	Euler_z	aVal_x	aVal_y	aVal_z
1	1	1	850	-0.05	-0.03	-0.46	46	-3	-5136
1	1	2	850	-0.09	-0.04	-0.48	68	28	-3756
1	1	3	850	-0.13	-0.04	-0.51	80	50	-2746
1	1	4	850	-0.16	-0.05	-0.54	81	64	-2009
1	1	5	850	-0.20	-0.05	-0.56	90	77	-1463
...
1	1	596	685	-28.20	-21.82	-35.37	-175	-1103	166

1	1	597	685	-28.22	-21.79	-35.30	-157	-1101	166
1	1	598	685	-28.23	-21.77	-35.21	-156	-1093	163
1	1	599	686	-28.25	-21.75	-35.13	-157	-1089	162
1	1	600	685	-28.26	-21.73	-35.05	-166	-1078	157

Fonte: Da autora (2021)

3.3 CONFECÇÃO DO MODELO COMPUTACIONAL DO MÓDULO PROCESSADOR

O modelo foi desenvolvido por meio da aplicação web *Jupyter Notebook*, a fim de que os dados e os resultados das análises fossem facilmente visualizados. As principais bibliotecas utilizadas foram o *NumPy*, *TensorFlow 2.0* e o *Keras*. Abaixo seguem as principais funcionalidades das mesmas:

- **NumPy:** Fornece um extenso conjunto de funções que auxiliam na execução de cálculos matemáticos em *arrays* multidimensionais. Os *Arrays NumPy* foram utilizados para armazenar os dados de treinamento e os parâmetros do modelo.
- **TensorFlow 2.0:** Esta biblioteca foi criada pela equipe *Google Brain* e reúne uma série de algoritmos e modelos de *machine learning* e aprendizagem profunda, as quais foram essenciais para a confecção da rede neural recorrente.
- **Keras:** Desenvolvida com o foco de possibilitar rápida experimentação, o *Keras* é uma API (Interface de Programação de Aplicativo) de redes neurais de alto nível capaz de rodar em cima do *TensorFlow*.

4 RESULTADOS E DISCUSSÕES

Neste capítulo será demonstrada a análise e segmentação dos dados obtidos através do módulo coletor. Em sequência é demonstrada a arquitetura final da rede neural responsável pela classificação dos registros e, como a previsão, foi indicada em forma de áudio.

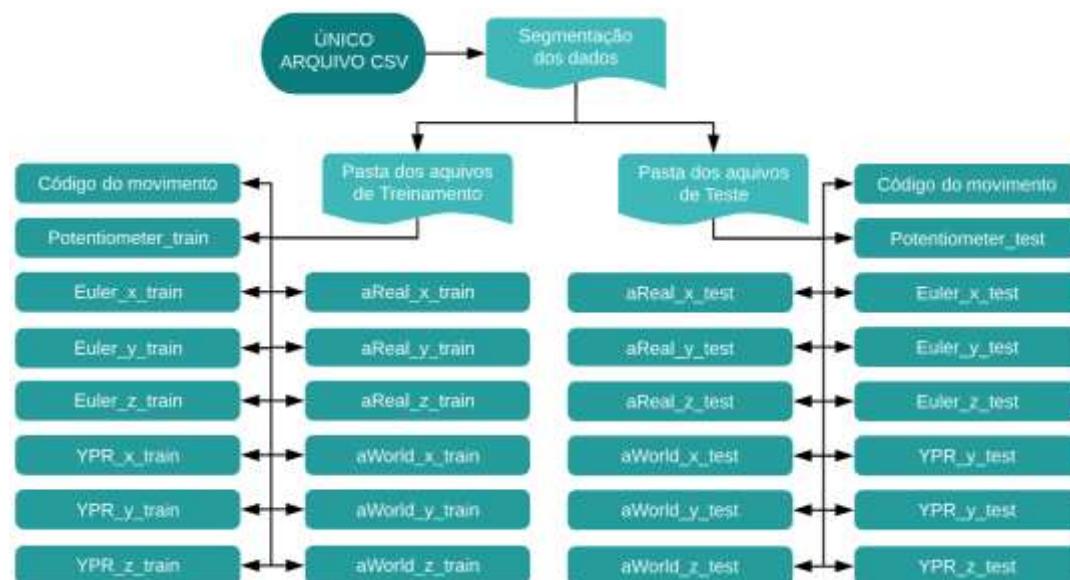
4.1 ORGANIZAÇÃO DOS DADOS

A partir da coleta de 814 amostras, igualmente repartidas entre 407 registros de movimentação referentes às expressões “oi” e “tudo bem?”, foi realizada a segmentação dos dados adquiridos em arquivos CSV distintos, a fim de facilitar a

alimentação da rede neural. Desse modo, a ordem das linhas de registro em cada um dos arquivos representava a mesma gravação de atividade. Neste momento, as gravações foram repartidas entre dados destinados ao treinamento da rede neural (70%) e dados reservados para teste (30%), visto que a utilização dos mesmos valores para estas duas etapas pode resultar em uma falsa indicação de eficácia.

A divisão dos dados das gravações está indicada pela Fig. 10, na qual são demonstradas as variáveis disponíveis para o treinamento da rede neural. As variáveis são iniciadas em “Potenciometer”, “Euler”, “YPR”, “aReal” e “aWord” o que simboliza, respectivamente, se os dados são referentes ao potenciômetro, os ângulos de Euler, os componentes de arfagem, guinada ou rolagem (*Pitch, Yaw e Roll*), os componentes da aceleração com a gravidade removida ou os componentes da aceleração com a remoção da gravidade e referenciados com a orientação inicial.

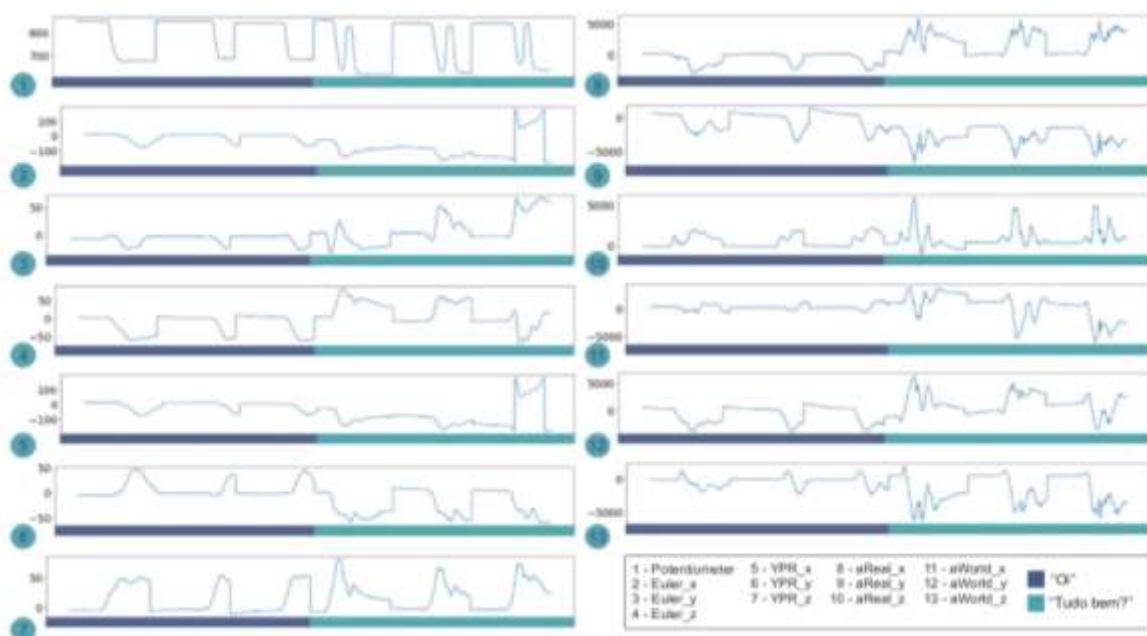
Figura 10: Descrição da segmentação dos dados obtidos



Fonte: Da autora (2021)

Com o intuito de comparar as gravações adquiridas, foi utilizada a função *pyplot* da biblioteca *matplotlib* para confecção de 13 gráficos, indicados pela Fig.11, referentes a seis gravações de movimentos. Com isto, foi possível observar padrões existentes na execução destes movimentos, assim como a presença de possíveis interferências no caso da sexta amostra para as variáveis Euler_x, Euler_y e YPR_x, as quais expressaram valores destoantes dos apresentados nas outras gravações para o mesmo movimento.

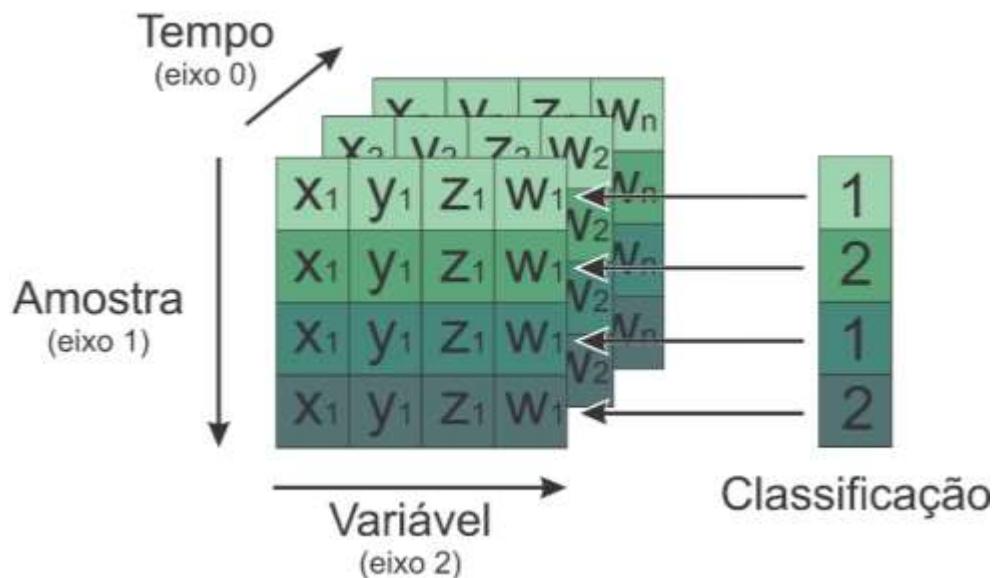
Figura 11: Análise dos registros de movimentações para diferentes expressões



Fonte: Da autora (2021)

Dessa forma, os dados de entrada para o treinamento e teste da rede neural recorrente foram adicionados ao programa em dois array *NumPy* tridimensional distintos, cujas dimensões são amostra, tempo e variável. Também foram criados dois array unidimensionais referentes a classificação real das amostras, as estruturas de ambos tipos de armazenamento de variáveis são demonstradas com a Fig. 12.

Figura 12: Estrutura dos arrays utilizados para armazenamento das variáveis.

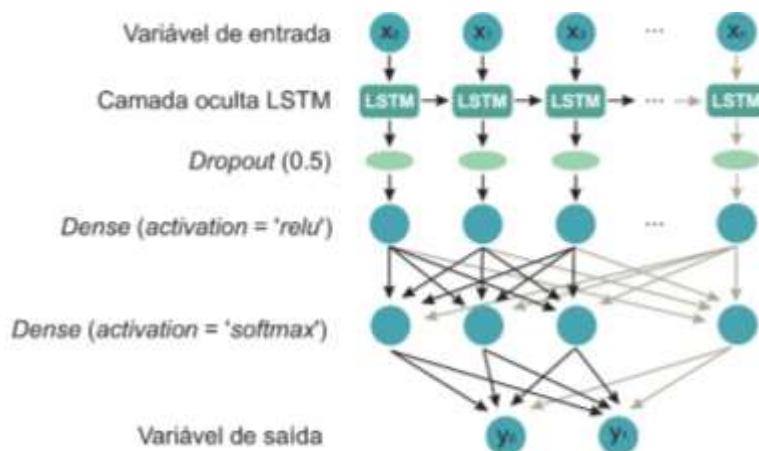


Fonte: Da autora (2021)

4.2 ESTRUTURA DA REDE NEURAL RESULTANTE

Para a confecção da rede neural utilizou-se o modelo *Sequential* Keras de Interface de Programação de Aplicação (API), por conta da sua praticidade de uma simples lista de camadas da rede neural, associado ao fator de que a limitação de entrada e saída únicas não interferem no resultado esperado. A organização da arquitetura final da rede neural para classificação dos movimentos é demonstrada pela Fig. 13.

Figura 13: Fluxograma da arquitetura final



Fonte: Da autora (2021)

Com relação à ordem das camadas escolhidas para composição da rede neural, primeiramente, foi adicionado a camada oculta de arquitetura LSTM, a fim de que o modelo aprenda qual informação deverá ser armazenada à memória de longo prazo e de qual deve se desfazer. Em sequência adicionou-se a camada *Dropout* que atua durante o período de treinamento com a finalidade de prevenir *overfitting*, ou seja, um sobre ajuste do modelo aos dados utilizados para o treinamento, sem generalizar adequadamente para a compressão de novos dados.

Foram adicionadas duas camadas *Dense*. A primeira camada é oculta e possui realiza a interpretação dos atributos extraídos pela camada oculta LSTM, através da função de ativação *relu* (*rectified linear unit activation function*). Por fim, a última camada *Dense* é responsável por realizar as previsões ao informar o total de saídas e a função de ativação *softmax*, a qual é descrita pela documentação oficial do

Keras como comumente aplicada na última camada de redes neurais de classificação, visto que o resultado pode ser interpretado como uma distribuição de probabilidade. Os argumentos escolhidos para a compilação do modelo Keras foram: perda = *categorical_crossentropy*, otimizador = Adam e métrica = *accuracy*. A perda se refere a quanto longe as previsões estão dos valores reais e, com isto a perda de entropia cruzada é utilizada quando há dois ou mais classes. O otimizador Adam é um método de estocástico computacionalmente eficiente e com poucos requisitos de memória. Com o uso da métrica de precisão são criadas duas variáveis locais que são utilizadas para indicar a frequência em que o valor previsto corresponde ao resultado real. A Tab. 2 exibe estes e demais parâmetros utilizados para a confecção da rede neural.

Tabela 2: Parâmetros para construção da rede neural

Parâmetro	Valor
Épocas	15
Tamanho do Lote	64
LSTM (<i>units</i>)	100
<i>Dropout</i>	0.5
Dense oculta (<i>units</i>)	100
Dense oculta (ativação)	ReLU
Dense saída (<i>units</i>)	2 (Número de saídas)
Dense saída (ativação)	Softmax
Perda	'categorical_crossentropy'
Otimizador	Adam
Métrica	Acurácia

Fonte: Da autora (2021)

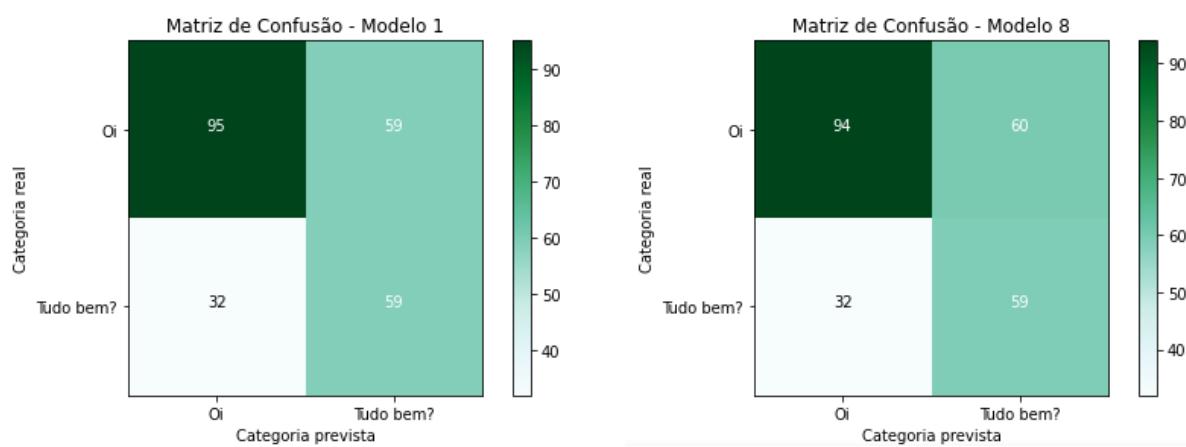
4.3 ANÁLISE DA PRECISÃO

Dada a natureza estocástica da rede neural confeccionada, a performance não poderia ser julgada por uma avaliação única, visto que foram seguidas sequências de bits síncronas a um certo sinal *clock*, que são gerados por processos aleatórios independentes para cada bit.

Dito isso, a avaliação do modelo foi repetida dez vezes e, após a finalização dos testes, a média da pontuação de precisão foi de 62,816% (+/-0.339). Foram

geradas duas matrizes de confusão a fim de analisar os modelos de maior e menor precisão indicadas pela Fig. 14, nos quais são analisadas as ocorrências durante a etapa de testes. É possível observar que a precisão para detecção do gesto da palavra “oi” está acima de 61%. Contudo, a expressão “tudo bem?” alcançou apenas próximo de 65% de exatidão em classificações para a avaliação utilizada para análise.

Figura 14: Classificação da matriz de confusão utilizando o modelo proposto



Fonte: Da autora (2021)

Ao observar artigos que estudam o reconhecimento de atividades humanas (*Human Activity Recognition* - HAR), através de redes neurais artificiais e *Deep Learning*, são encontrados valores de precisão superiores a 90%. Dentre as características principais destes trabalhos podem ser citadas:

- **Maior número de camadas ocultas de arquitetura LSTM:** o modelo de rede neural de longo-curto prazo para análise HAR através de acelerômetro e giroscópio, apresentados por Tüfek e Özka (2019), exibe precisão de 98% ao utilizar três camadas do tipo LSTM.
- **Uso de camadas de arquitetura convolucional e LSTM:** o estudo apresentado por Xia, Huang e Wang (2020) aponta uma precisão de 95.8% na identificação dos movimentos registrados através do acelerômetro de um smartphone, disponibilizado pelo dataset público UCI-HAR.
- **Uso de camadas convolucionais e bidirecionais LSTM:** conforme alcançado por Nafea, Abdul, Muhammad e Alsulaiman (2021), com a utilização de camadas ocultas convolucionais e bidirecionais LSTM, foi alcançada uma precisão de 97.05% para a previsão das atividades presentes no dataset UCI-HAR.

Outro fator passível de auxiliar no aumento da precisão do modelo é o registro de um maior número de amostras utilizadas na etapa de treinamento. Em razão de que para volumes de dados considerados pequenos, abaixo de quatro milhões de registros, a precisão de reconhecimento para modelos de *Deep Learning* aumenta rapidamente, conforme cresce a quantidade de dados disponíveis para treinamento (CHEN et al., 2017).

4.3 REPRODUÇÃO DOS RESULTADOS EM ÁUDIO

Após a classificação, a amostra de teste selecionada foi convertida em arquivo de áudio, através da biblioteca gTTS (Google *Text-to-Speech*) o qual é reproduzido com o auxílio do módulo *playsound* os textos “oi” ou “tudo bem?”, como indica a Fig. 15. Contudo, dada a precisão da classificação citada anteriormente, ocorreram testes em que o conteúdo do áudio não representava o real movimento registrado.

Figura 15: Código para reprodução do texto conforme a classificação da atividade



```
In [9]: #corregra o modelo com menor classificação
file = "FLIBRASrecognizer.h5"
model = tf.keras.models.load_model(file)

#corregra os dados de movimento (X) e a classificação dos mesmos (y)
trainX, trainy, testX, testy = load_dataset()

#realiza a classificação das amostras de teste
predictions = model.predict(testX, batch_size=10, verbose=0)

#ajusta o array das previsões e suas classificações reais
rounded_predictions = np.argmax(predictions, axis=1)
rounded_labels = np.argmax(testy, axis=1)

#Mostra a amostra i para realizar a classificação
sample = rounded_predictions[i]
print("A classificação prevista foi: ", sample)
print("A classificação real foi: ", rounded_labels[i])

In [10]: # 0 = "Ol", 1 = "Tudo bem?"
Speech = ["Ol", "Tudo bem!"]
var = gTTS(Speech[sample], lang = "pt")

file = "TextToSpeech.mp3"

if os.path.exists(file):
    os.remove(file)
    var.save(file)
else:
    var.save(file)

playsound(file)
```

Fonte: Da autora (2021)

5 CONCLUSÕES

Este trabalho teve a proposta de elaborar um protótipo que auxiliasse na comunicação entre surdos e ouvintes no sentido de permitir que o primeiro grupo seja escutado. Dessa forma, implementou-se a robótica assistiva, manufaturada através da impressão 3D, para que as informações relacionadas às configurações de mão exercidas sejam registradas pelo módulo coletor e reconhecidas por um módulo

processador separado possuindo de uma rede neural treinada para classificação dessas informações.

O módulo coletor registrou informações da compressão do dedo indicador e da orientação espacial da mão conforme os componentes de aceleração e ângulos resultantes dos movimentos após do tratamento com um filtro complementar, resultado da associação dos sinais do giroscópio e do acelerômetro, presentes no sensor MPU-6050. A rede neural resultante possui uma camada oculta de arquitetura LSTM responsável por identificar quais características são necessárias de serem armazenadas a longo prazo e quais deviam ser descartadas.

A análise dos dados adquiridos através do sistema mecânico fabricado com manufatura aditiva permitiu contemplar padrões realizados durante a movimentação das mãos durante a fala em LIBRAS. Desta forma a rede neural artificial desenvolvida possui caráter científico a fim de estudar a identificação dos padrões presentes nos registros. Como resultado foi atingida uma precisão de 62,816% (+/-0.339) na classificação de amostras em análise entre os dois gestos utilizados durante a etapa de treinamento.

As repercussões do presente trabalho indicam que a de confecção de um protótipo mecatrônico pode ser aplicado como iniciativa de módulo coletor para registro dos gestos. Também foi possível observar através do modelo computacional produzido e dos estudos analisados a aplicabilidade de redes neurais na classificação de movimentos humanos. Por fim, ao expressar os resultados obtidos em forma de áudio o estudo se demonstra alcançar o objetivo de trazer uma iniciativa de auxílio na comunicação entre surdos e ouvintes.

As principais melhorias a serem implementadas em trabalhos futuros são o aumento do vocabulário e o aprimoramento da precisão do modelo. Conforme linhas de pesquisas indicam este segundo aperfeiçoamento pode ser alcançado através da aquisição de um volume maior de dados utilizados na etapa de treinamento ou alterações na estrutura escolhida para confecção da rede neural.

REFERÊNCIAS

AMBROŽ, Miha. Raspberry Pi as a low-cost data acquisition system for human powered vehicles. **Measurement**, v. 100, p. 7-18, mar. 2017. Elsevier BV.
<http://dx.doi.org/10.1016/j.measurement.2016.12.037>.

ARDUINO. **Arduino Nano**. Disponível em: <https://store.arduino.cc/usa/arduino-nano>. Acesso em: 14 jun. 2020.

BANCALEIRO, Cláudia. **Tablet Uni permite que surdos comuniquem sem limitações**. 2014. Disponível em: <https://www.publico.pt/2014/10/24/tecnologia/noticia/tablet-uni-permite-que-surdos-comuniquem-sem-limitacoes-1673901>. Acesso em: 14 jun. 2020.

CANIZARES, Alejandro; PAZOS, Jean; BENITEZ, Diego. On the use of 3D printing technology towards the development of a low-cost robotic prosthetic arm. **2017 IEEE International Autumn Meeting On Power, Electronics And Computing (ropec)**, [s.l.], p. 1-6, nov. 2017. IEEE. <http://dx.doi.org/10.1109/ropec.2017.8261579>.

CHEN, Haipeng; XIONG, Fuhai; WU, Dihong; ZHENG, Lingxiang; PENG, Ao; HONG, Xuemin; TANG, Biyu; LU, Hai; SHI, Haibin; ZHENG, Huiru. Assessing impacts of data volume and data set balance in using deep learning approach to human activity recognition. **2017 IEEE International Conference On Bioinformatics And Biomedicine (Bibm)**, Kansas, p. 1-6, nov. 2017. IEEE. <http://dx.doi.org/10.1109/bibm.2017.8217821>.

COLELLA, Riccardo; CATARINUCCI, Luca. Wearable UHF RFID Sensor-Tag Based on Customized 3D-Printed Antenna Substrates. **IEEE Sensors Journal**, v. 18, n. 21, p. 8789-8795, 1 nov. 2018. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/jsen.2018.2867597>.

COLLADO, Maria Laide dos Santos. **A participação política do surdo: há um poder em suas mãos**. 2016. 128 f. Dissertação (Mestrado) - Curso de Educação, Universidade Nove de Julho – Uninove, São Paulo, 2016.

ELETROÔNICA, Baú da. **Módulo Bluetooth HC-05**. Disponível em: <https://www.baudaelectronica.com.br/modulo-bluetooth-hc-05.html>. Acesso em: 14 jun. 2020.

EUGENIO, Kassio Janielson da Silva. **IMPLEMENTAÇÃO DE ARQUITETURA DE HARDWARE E SOFTWARE PARA A ÓRTESE ATIVA PARA OS MEMBROS INFERIORES ORTHOLEG**. 2016. 44 f. TCC (Graduação) - Curso de Engenharia Mecatrônica, Universidade Federal do Rio Grande do Norte, Natal, 2016.

FLORÊNCIO, Ayrton Galindo Bernardino. **Sistema de estabilização de vídeo baseado em acelerômetro, filtragem robusta e algoritmo de busca de três etapas**. 2015. 106 f. Tese (Doutorado) - Curso de Ciências, Programa de Engenharia Elétrica, Escola de Engenharia de São Carlos da Universidade de São Paulo, São Carlos, 2015.

FOUNDATION, Raspberry Pi. **Raspberry Pi 3 Model B**. Disponível em: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>. Acesso em: 14 jun. 2020.

FRANCZAK, Douglas Diogo. **MYMAPPER: plataforma de mapeamento de ambientes 3d**. 2016. 87 f. TCC (Graduação) - Curso de Mecatrônica Industrial,

Departamentos Acadêmicos de Eletrônica e Mecânica, Universidade Tecnológica Federal do Paraná – Ufpr, Curitiba, 2016.

FRIZON, Lucas JosÉ; PECCIN, Vinícius Berndsen; JINBO, Maro. **Estimação dinâmica de ângulo baseado em giroscópio e acelerômetro com filtro de Kalman**. 2015. 6 f. TCC (Graduação) - Curso de Engenharia de Controle e Automação, Depto. Engenharia de Controle e Automação, Instituto Federal de Santa Catarina, Chapecó, 2015.

GANDRA, Alana. **País tem 10,7 milhões de pessoas com deficiência auditiva, diz estudo**. Agência Brasil, Rio de Janeiro, 13 de out. de 2019. Disponível em: <https://agenciabrasil.ebc.com.br/geral/noticia/2019-10/brasil-tem-107-milhoes-de-deficientes-auditivos-diz-estudo>. Acesso em: 28 jun. 2021.

GRUS, Joel. **Data Science do Zero**: primeiras regras com o Python. Rio de Janeiro: Alta Books, 2016. 315 p.

GUERTIN, Steven M. Guideline for Single-Event Effect (SEE) testing of System on a Chip (SOC) devices. **National Aeronautics And Space Administration**, Pasadena, p. 1-130, 1 fev. 2018.

IWANIEC, Marek et al. Development of Vibration Spectrum Analyzer Using the Raspberry Pi Microcomputer and 3-Axis Digital MEMS Accelerometer ADXL345. In: XIIITH International Conference on Perspective Technologies And Methods In Mems Design (MEMSTECH), 13, 2017, Polyana. **Conferência**. Ukraine: Ieee, 2017. p. 25-29.

JESUS, Vinicius Souza de; SAMYN, Leandro Marques; SAMYN, Leandro Marques. Desenvolvimento de uma Interface para a Comunicação LIBRAS – Língua Portuguesa. In: MOSTRA NACIONAL DE ROBÓTICA (MNR), 2015, Uberlândia. **Conferência**. Uberlândia: Researchgate, 2015. p. 1-6.

KAMAL, Raj. **Embedded Systems - SoC, IoT, AI and Real-Time Systems**. 4. ed. Nova Iorque: Mcgraw-hill Education, 2020. 598 p.

LU, Yu-sheng; WANG, Hsuan-wen; LIU, Sheng-hao. An integrated accelerometer for dynamic motion systems. **Measurement**, v. 125, p. 471-475, set. 2018. Elsevier BV. <http://dx.doi.org/10.1016/j.measurement.2018.05.019>.

LUBBADEH, Mana; SHASHTARI, Aya. **Smart Deaf-Mute Glove**. 2017. 52 f. TCC (Graduação) - Curso de Engenharia da Computação, An-najah National University, Nablus, 2017.

NAFEA, Ohoud; ABDUL, Wadood; MUHAMMAD, Ghulam; ALSULAIMAN, Mansour. Sensor-Based Human Activity Recognition with Spatio-Temporal Deep Learning. **Sensors**, [S.L.], v. 21, n. 6, p. 2141, 18 mar. 2021. MDPI AG. <http://dx.doi.org/10.3390/s21062141>.

O'HARE, Ryan. **'SignAloud' gloves translate sign language gestures into spoken English**. 2020. Disponível em: <https://www.dailymail.co.uk/sciencetech/article-3557362/SignAloud-gloves-translate-sign-language-movements-spoken-English.html>. Acesso em: 14 jun. 2020.

OZ, Cemil; LEU, Ming C.. American Sign Language word recognition with a sensory glove using artificial neural networks. **Engineering Applications Of Artificial Intelligence**, v. 24, n. 7, p. 1204-1213, out. 2011. Elsevier BV.
<http://dx.doi.org/10.1016/j.engappai.2011.06.015>.

PARK, Eun-jung; SEHDEV, Namita; FROGOSO, Roger. American Sign Language Translator using Gesture Recognition. **The City College Of New York grove School Of engineering**. New York, p. 1-15. set. 2016.

RICHARDSON, Matt; WALLACE, Shawn. **Primeiros Passos com o Raspberry PI**. São Paulo: Novatec, 2013. 192 p.

SILVA, Fernando Ginez da. **Reconhecimento de movimentos humanos utilizando um acelerômetro e inteligência computacional**. 2013. 105 f. Dissertação (Mestrado) - Curso de Engenharia, Escola Politécnica da Universidade de São Paulo, São Paulo, 2013.

SOARES, Fábio M.; SOUZA, Alan M. F.. **Neural Network Programming with Java: Unleash the power of neural network by implementing professional java code**. Birmingham: Packt Publishing, 2016. 244 p.

SU, Wenjing et al. 3D printed wearable flexible SIW and microfluidics sensors for Internet of Things and smart health applications. **2017 Ieee Mtt-s International Microwave Symposium (ims)**, p. 1-4, jun. 2017. IEEE.
<http://dx.doi.org/10.1109/mwsym.2017.8058621>.

TEAGUE, Caitlin N. et al. A Wearable, Multimodal Sensing System to Monitor Knee Joint Health. **IEEE Sensors Journal**, p. 1-1, 2020. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/jsen.2020.2994552>.

TEVES, André da Costa. **Otimização de Acelerômetros MEMS Eletroestáticos de Alto Desempenho**. 2013. 128 f. Dissertação (Mestrado) - Curso de Engenharia, Escola Politécnica da Universidade de São Paulo, São Paulo, 2013.

TOLEDO JÚNIOR, Emerson Galdino. **Acelerômetro sem fio de baixo custo para monitoramento dinâmico estrutural**. 2018. 100 f. Dissertação (Mestrado) - Curso de Engenharia Civil, Universidade Federal de Juiz de Fora, Juiz de Fora, 2018.

TUFEK, Nilay; OZKAYA, Ozen. A Comparative Research on Human Activity Recognition Using Deep Learning. **2019 27Th Signal Processing And Communications Applications Conference (Siu)**, p. 1-4, abr. 2019. IEEE.
<http://dx.doi.org/10.1109/siu.2019.8806395>.

VICENZI, Gustavo. **Análise biomecânica-computacional de movimentos do ombro por meio do uso de acelerômetro**. 2019. 120 f. Dissertação (Mestrado) - Curso de Ciências Aplicadas à Saúde, Universidade Estadual do Oeste do Paraná, Francisco Beltrão, 2019.

XIA, Kun; HUANG, Jianguang; WANG, Hanyu. LSTM-CNN Architecture for Human Activity Recognition. **Ieee Access**, v. 8, p. 56855-56866, 2020. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/access.2020.2982225>.



YOON, Jinsu et. al. Monolithically 3D-printed pressure sensors for application in electronic skin and healthcare monitoring. **2017 IEEE International Electron Devices Meeting (IEDM)**, p. 1-4, dez. 2017. IEEE.
<http://dx.doi.org/10.1109/iedm.2017.8268525>.

YOUNBIONIC. **Another Hand**. Disponível em:
<https://www.youbionic.com/anotherhand>. Acesso em: 14 jun. 2020.