

DESENVOLVIMENTO DE UMA INTERFACE BASEADA EM GESTOS DAS MÃOS PARA APRESENTAÇÕES

Gabriel Mendes Camargo de Barros
Graduando em Sistemas de Informação – Uni-FACEF
gmendescamargodebarros@gmail.com

Pedro Antônio Rodrigues Faleiros
Graduando em Sistemas de Informação – Uni-FACEF
pedro.arf00@gmail.com

Jaqueline Brigladori Pugliesi
Doutora em Ciência da Computação – USP São Carlos
jbpugliesi@gmail.com

Resumo

Este trabalho apresenta o desenvolvimento de uma interface baseada em gestos das mãos para controle de apresentações de slides. O sistema é composto por uma luva equipada com sensores de acelerômetro e giroscópio (MPU6050), integrados a um microcontrolador ESP8266. Os dados capturados são transmitidos via protocolo MQTT para um sistema de inteligência artificial. A implementação utilizou técnicas de aprendizado profundo, especificamente um modelo Transformer, para processar e classificar os dados dos sensores em tempo real. Os testes realizados mostraram uma alta precisão no reconhecimento de gestos. O sistema proposto oferece uma alternativa intuitiva e natural aos métodos convencionais de controle de apresentações, como teclado e mouse, destacando o potencial das interfaces gestuais na melhoria da interação humano-computador.

Palavras-chave: Reconhecimento de Gestos. Inteligência Artificial. Acelerômetros. Controle por Gestos. MQTT. NodeMCU ESP8266.

Abstract

This work presents the development of an interface based on hand gestures for controlling slideshows. The system consists of a glove equipped with accelerometer and gyroscope sensors (MPU6050), integrated with an ESP8266 microcontroller. The captured data is transmitted via MQTT protocol to an artificial intelligence system. The implementation used deep learning techniques, specifically a Transformer model, to process and classify sensor data in real-time. The tests performed demonstrated high accuracy in gesture recognition. The proposed system offers an intuitive and natural alternative to conventional presentation control methods, such as keyboard and mouse, highlighting the potential of gestural interfaces in improving human-computer interaction.

Keywords: Gesture Recognition. Artificial Intelligence. Accelerometers. Gesture Control. MQTT. NodeMCU ESP8266.

1 Introdução

No campo das interfaces homem-máquina, o reconhecimento gestual, especialmente por meio de sensores de acelerometria, tem se destacado como uma alternativa promissora para oferecer uma experiência de interação mais intuitiva e natural. Essa tecnologia permite que comandos sejam executados mediante movimentos das mãos, explorando novas possibilidades de controle e interação com dispositivos tecnológicos.

O desenvolvimento de uma interface gestual das mãos, que integra sensores de acelerômetro com Inteligência Artificial (IA), surge como uma solução para otimizar a interação dos usuários com sistemas de apresentação. A aplicação de IA aprimora a precisão e a adaptabilidade da interface, tornando o reconhecimento de gestos mais eficiente e personalizado. Tal sistema pode substituir ou complementar métodos tradicionais de controle, como teclado e mouse, sendo especialmente útil em contextos como simulações para ambientes de realidade virtual e aumentada.

Este trabalho tem como objetivo o desenvolvimento de um protótipo de luva com reconhecimento gestual, visando explorar uma aplicação prática do potencial desta abordagem. A proposta se concentra na integração de Inteligência Artificial com sensores de acelerômetro para criar uma solução que transforme a interação com tecnologias em contextos específicos, como apresentações de slides. Ao combinar esses elementos, o trabalho visa não apenas mostrar a eficácia e a versatilidade do reconhecimento gestual, mas também oferecer uma nova abordagem para a interação com dispositivos tecnológicos, destacando seu impacto potencial na maneira como se conduz e experimenta apresentações e outras aplicações.

Este projeto possui um caráter exploratório e prático, voltado para a construção de um protótipo de luva com reconhecimento gestual de mãos, tendo como base partes do projeto *Wave Glove*, desenvolvido por Matej Králik (Králik, 2020). O desenvolvimento do hardware foi realizado no Arduino IDE, que permite a integração dos sensores de acelerômetro para capturar e interpretar os gestos. Foi utilizado o protocolo *Message Queuing Telemetry Transport* (MQTT) para a comunicação da luva com o computador e a linguagem de programação Python para o desenvolvimento da camada de software responsável pelo recebimento e tratamento dos dados, que inclui o treinamento e a execução do modelo de reconhecimento dos gestos e o PyAutoGUI, uma biblioteca responsável por executar comandos de mouse e teclado.

Além disso, o projeto envolve o uso de Inteligência Artificial para o treinamento e reconhecimento dos gestos, sendo que para o treinamento foi escolhido o modelo *transformers*, que é um método de treinamento de *deep learning*. A aplicação prática é a demonstração do controle de apresentações de slides, no qual o protótipo permitirá a navegação pelos slides apenas com o uso de movimentos gestuais.

2 Referencial Teórico

Nesta seção, serão discutidos tecnologias, instrumentos e princípios empregados ao longo do processo de criação da aplicação, com o propósito de esclarecer ao leitor suas características mais relevantes.

2.1 Arduino

O Arduino é uma plataforma de prototipagem eletrônica de código aberto, desenvolvida por cinco pesquisadores: Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis, em 2005, na Itália (Makiyama, 2022). Segundo Banzi & Shiloh (2014), projetado para facilitar a criação de projetos eletrônicos interativos, o Arduino é um pequeno computador em uma placa, equipado com um microcontrolador e pinos que permitem conectar diversos componentes eletrônicos, como sensores e motores. Sua linguagem de programação é baseada em C/C++, possibilitando a integração modular de sensores, atuadores e módulos de comunicação, permitindo a criação de dispositivos personalizados que atendem a requisitos específicos.

No contexto da *Internet of Things* (IoT), o Arduino desempenha um papel fundamental ao possibilitar a criação de sistemas conectados e inteligentes. Com sua interface simples e suporte a diversos módulos de comunicação, como o ESP8266, permite que dispositivos físicos se conectem à internet de forma acessível e eficiente. Isso possibilita o monitoramento remoto de variáveis ambientais e o controle de dispositivos em tempo real, essenciais para aplicações como automação residencial, gerenciamento de infraestrutura urbana e controle industrial (Hughes, 2016).

De acordo com Vimali et al. (2021) permitir que os usuários comuniquem comandos por meio de movimentos das mãos, visto que tradicionalmente esses gestos são utilizados como entradas para sistemas robóticos, proporciona uma forma intuitiva de controle. A abordagem visa explorar a flexibilidade e a simplicidade do Arduino para criar um dispositivo de entrada de gestos acessível, ampliando seu potencial de aplicação em diversas soluções práticas.

2.2 Sensores de Acelerômetro

Os acelerômetros são sensores projetados para medir a aceleração de um objeto em movimento, baseando-se em princípios físicos que utilizam forças inerciais para identificar variações de velocidade ou direção. Compostos por elementos como massas suspensas ou cristais piezoelétricos, esses sensores convertem a aceleração em sinais elétricos proporcionais (Guimarães, 2018). São utilizados em dispositivos como smartphones e tablets, detectam a orientação e permitem a rotação automática da tela, também são essenciais em *wearables*, que são tecnologias vestíveis, como relógios inteligentes e rastreadores de atividades físicas, oferecendo precisão no monitoramento de passos, sono e movimentos (Aero, 2023). Sua aplicação em realidade aumentada e virtual mostra o potencial dessa tecnologia em interações mais avançadas.

A tecnologia que viabiliza os acelerômetros modernos está baseada em sistemas *Microelectromechanical Systems* (MEMS). Esses sistemas miniaturizados combinam sensores, atuadores e circuitos, permitindo medições precisas. Segundo Hsu (2008), acelerômetros utilizam um sistema massa-mola microscópico, no qual a aceleração provoca deformações que geram sinais elétricos. Graças à sua precisão e confiabilidade, os MEMS se tornaram fundamentais em áreas como automação, medicina e aeronáutica, além de sua presença no cotidiano. A capacidade de analisar movimentos e forças de forma eficiente amplia sua utilidade, desde o controle por gestos até o monitoramento de vibrações em estruturas.

2.3 Inteligência Artificial Aplicada a Reconhecimento de Gestos

A Inteligência Artificial emergiu como uma das tecnologias mais transformadoras do século XXI, impactando uma ampla gama de setores, desde a automação industrial até as interações homem-máquina. Em particular, no campo do reconhecimento de gestos, a IA tem desempenhado um papel fundamental, permitindo que sistemas computacionais interpretem e respondam a movimentos humanos de maneira intuitiva e eficaz. Essa tecnologia torna as interfaces de usuário mais naturais e fluidas, especialmente em sistemas de controle, nos quais a interpretação precisa de gestos pode revolucionar a experiência do usuário. Conforme discutido por Mitra e Acharya (2007), a habilidade de reconhecer gestos com precisão é crucial para a evolução das interfaces interativas, destacando a importância da IA nesse processo.

A IA, por definição, refere-se a sistemas computacionais projetados para executar tarefas que normalmente requerem inteligência humana. Esses sistemas dependem de algoritmos sofisticados para aprender com dados, fazer previsões e tomar decisões. Um dos subcampos mais promissores da IA é o aprendizado de máquina, que envolve o desenvolvimento de algoritmos capazes de aprender a partir de dados e melhorar seu desempenho ao longo do tempo (Goodfellow; Bengio; Courville, 2016). Um exemplo notável são as redes neurais artificiais, inspiradas no funcionamento do cérebro humano, que têm sido amplamente utilizadas para tarefas de reconhecimento de padrões, como o reconhecimento de gestos (LeCun; Bengio; Hinton, 2015). O *deep learning* é essencial para que os sistemas realizem o reconhecimento de gestos de forma precisa e em tempo real.

Nos últimos anos, o avanço do *deep learning* tem sido notável, com aplicações que se estendem a diversas áreas de pesquisa. Um dos campos que mais se beneficiou dessa evolução é o *Human Activity Recognition* (HAR). Králik (2020) observa que essa tecnologia, caracterizada por sua capacidade de aprender representações de dados de maneira hierárquica, tem se mostrado extremamente eficaz na extração de características complexas e abstratas. Ao contrário dos métodos tradicionais, que dependem da engenharia manual de características, ele é capaz de aprender automaticamente a partir dos dados, eliminando a necessidade de intervenção humana direta no processo de modelagem. Isso torna o *deep learning* uma ferramenta valiosa em aplicações complexas como o HAR, em que os padrões a serem identificados são intrincados e dinâmicos.

Outro avanço significativo no campo da IA é a arquitetura *Transformer*, proposta por Vaswani et al. (2017), que transformou o *Natural Language Processing* (NLP). Diferente das abordagens tradicionais baseadas em redes neurais recorrentes, como as *Long Short-Term Memory* (LSTMs) e *Gated Recurrent Units* (GRUs), os *transformers* utilizam exclusivamente mecanismos de atenção para modelar dependências globais entre as entradas e saídas. Isso elimina a necessidade de processar dados de forma sequencial, permitindo uma paralelização muito maior durante o treinamento dos modelos. Como resultado, os *transformers* não apenas melhoram a eficiência computacional, mas também elevam a qualidade em tarefas como tradução automática, estabelecendo um novo padrão de excelência.

Além disso, a introdução dos *transformers* trouxe um novo patamar de desempenho ao campo da tradução automática, tornando o processo de treinamento

mais rápido e eficiente. Vaswani et al. (2017) mostram que a arquitetura *Transformer* pode atingir resultados expressivos em tradução de idiomas após apenas doze horas de treinamento em um conjunto de *Graphics Processing Units* (GPUs). Essa abordagem superou modelos anteriores ao lidar com dependências de longa distância nas sequências de entrada e saída, oferecendo uma solução robusta para desafios que antes eram difíceis de abordar com redes neurais tradicionais.

No contexto de HAR, o modelo de *transformers* é usado no campo do reconhecimento de gestos manuais utilizando sensores inerciais. Kralik (2020) propõe um protótipo de luva equipado com múltiplos sensores inerciais que, em conjunto com uma arquitetura baseada em *Transformers*, alcança resultados significativos na detecção e classificação de gestos complexos. O uso do mecanismo de autoatenção permite capturar com precisão movimentos distintos dos sensores em cada dedo, o que representa uma melhoria em relação a métodos que utilizam apenas um sensor.

2.4 Interface Baseada em Gestos

O controle por gestos pode ser definido como uma série de interações entre humanos e a interface de máquinas que permite a emissão de comandos, abordando novos tipos de dispositivos ou complementando os tradicionais como teclado e mouse. Ao interpretar gestos realizados com as mãos ou outras partes do corpo, essas interfaces oferecem uma forma mais natural e intuitiva de interação, visando superar as limitações dos métodos convencionais de entrada de dados, proporcionando uma experiência mais fluida e imersiva para o usuário na interface de uma aplicação (Medeiros, 2015).

Esses gestos envolvem movimentos expressivos e significativos do corpo, como dos dedos, mãos, braços, cabeça ou rosto, com a intenção de transmitir informações ou interagir com o ambiente. Segundo Mitra e Acharya (2007), os gestos podem ser vistos como uma técnica de compressão de informações, que são posteriormente reconstruídas pelo receptor, permitindo uma comunicação mais eficiente entre o usuário e o sistema. As formas de gestos variam amplamente, podendo ser estáticas, quando o usuário assume uma determinada pose, ou dinâmicas, envolvendo fases de movimento antes, durante e após o gesto principal. Além disso, o reconhecimento automático de gestos contínuos naturais requer segmentação temporal para identificar corretamente os pontos de início e término dos movimentos em termos de espaço e tempo, o que pode ser influenciado pelo contexto de gestos anteriores e subsequentes.

Nos últimos anos, a crescente demanda por interações mais imersivas e naturais tem impulsionado o avanço do controle por gestos, como a popularização do Kinect, dispositivo comercializado pela Microsoft no console Xbox a partir de uma tecnologia criada por um grupo de pesquisadores israelenses Zeev Zalevsky, Alexander Shpunt, Aviad Maizels, Javier Garcia. Essa tecnologia, que envolve uma combinação de sensores de profundidade e câmeras para capturar a forma e o movimento do corpo humano, representa um marco significativo nessa evolução e mostra a viabilidade comercial e técnica do reconhecimento de gestos em tempo real, possibilitando a interação com o sistema de entretenimento por meio de movimentos corporais, sem a necessidade de controles físicos (Zalevsky et al, 2007).

Diferente de outras tecnologias de reconhecimento de gestos, como o *Microsoft Kinect*, que usa uma combinação de câmeras RGB e sensores de profundidade, o *Leap Motion*, um dispositivo de controle por gestos lançado pela primeira vez em 2013 pela empresa *Leap Motion, Inc*, se destaca por utilizar um sensor infravermelho de alta precisão. Esse sensor é capaz de detectar o movimento das mãos e dedos no espaço tridimensional com uma precisão milimétrica e uma taxa de amostragem de até 200 quadros por segundo, permitindo interações extremamente detalhadas e naturais no ambiente virtual (Weichert et al., 2013).

3 Desenvolvimento

Nesta seção serão apresentados os métodos e as ferramentas utilizados para o desenvolvimento do protótipo de luva com reconhecimento de gestos aplicado ao controle de apresentações de slides. A estruturação cuidadosa dos componentes, desde os sensores até a programação de todas as partes do software, constitui a base fundamental para a criação de um sistema funcional de reconhecimento gestual. O protótipo é estruturado em três subsistemas principais, cada um desempenhando um papel fundamental no funcionamento do sistema: Luva, compõe a estrutura de hardware necessária para a coleta dos sensores; Comunicação, trata-se da configuração e infraestrutura para o envio dos dados para o computador; e Sistema de Reconhecimento dos Gestos, subsistema responsável por gravar os dados para treinamento do modelo de IA, pelo reconhecimento dos gestos em tempo real e execução dos comandos para cada gesto.

3.1 Protótipo

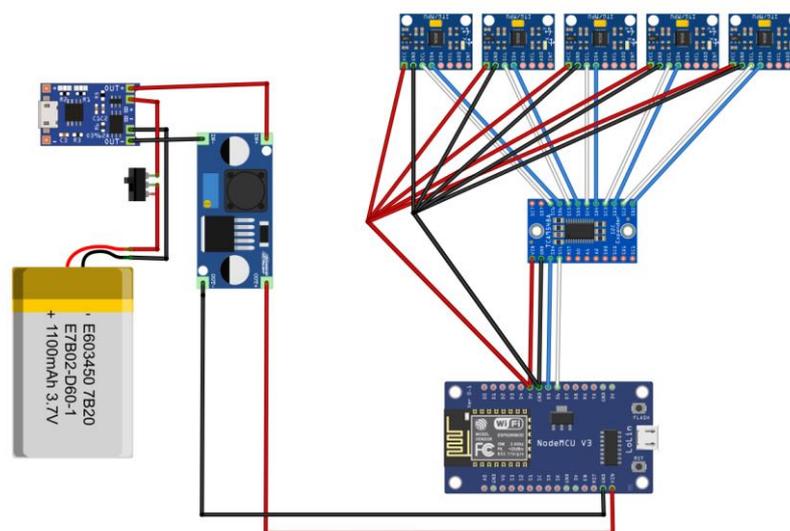
O primeiro subsistema consiste nos componentes diretamente integrados à luva, que são divididos em três partes, que incluem os sensores responsáveis pela captura dos gestos do usuário. Especificamente, são utilizadas *Inertial Measurement Units* (IMUs) MPU6050, que combinam acelerômetros e giroscópios para monitorar a movimentação em três eixos: x, y e z. Esses sensores estão estrategicamente posicionados nas falanges distais dos dedos, permitindo uma captura detalhada dos movimentos. O multiplexador TCA9548A, instalado no dorso da mão, é essencial para garantir que os principais pontos de movimento da mão sejam monitorados de forma eficiente, permitindo que múltiplos sensores sejam conectados simultaneamente e operem sem interferências. Com isso, o sistema é capaz de capturar com precisão a dinâmica dos gestos, proporcionando um mapeamento exato dos movimentos dos dedos e da mão.

Na segunda parte, o foco está na alimentação do sistema. Todo o sistema, incluindo o ESP8266 e os sensores MPU6050, é alimentado por uma bateria de íon-lítio (LiPo) de 3,7V e 500mA. O ESP8266, por sua vez, distribui a energia para os sensores, eliminando a necessidade de componentes adicionais de gerenciamento de energia. Para garantir a recarga da bateria, utiliza-se um carregador TP4056, que assegura o fornecimento contínuo de energia ao sistema.

A terceira parte envolve o microcontrolador, que desempenha um papel central na operação do protótipo. O ESP8266 desempenha o papel de capturar os dados das IMUs, os quais são os acelerômetros com giroscópio, por meio do protocolo de comunicação I2C (*Inter-Integrated Circuit*). Esses dados são então transmitidos

para um computador via Wi-Fi com o protocolo MQTT. Todo o processamento dos dados ocorre no computador, enquanto o microcontrolador tem a função exclusiva de requisitar os dados das IMUs, que estão conectadas em um multiplexador, conforme mostrado no esquema elétrico do protótipo da Figura 1.

Figura 1 – Esquema Elétrico do Protótipo.



Fonte: Os autores.

A utilização de cinco acelerômetros MPU 6050 oferece uma abordagem eficaz para capturar a movimentação dos cinco dedos da mão, como mostrado na Figura 2, o que possibilita o mapeamento com muita precisão de seus movimentos. Os valores dos eixos x, y e z provenientes dos giroscópios e acelerômetros são digitalizados pelo MPU 6050. Para isso, são utilizados seis conversores A/D (Analogico/Digital), sendo três para o acelerômetro e três para o giroscópio. Posteriormente, esses valores digitalizados são armazenados em registradores internos de 16 bits (Quirino; Girardi, 2019, p. 2).

Cada MPU 6050 possui um acelerômetro de três eixos, responsável por medir a aceleração linear nos eixos x, y e z, além de um giroscópio de três eixos que mede a aceleração angular sobre os eixos x, y e z obtendo assim uma boa noção de espaço tridimensional. O acelerômetro é capaz de medir acelerações nos três eixos separadamente até 16g (156,896 m/s²) e possui quatro faixas programáveis: 2g, 4g, 6g e 16g. A unidade g refere-se ao valor da aceleração gravitacional da Terra. O giroscópio é capaz de medir variações instantâneas nos três eixos separadamente com quatro faixas programáveis: 250°/s, 500°/s, 1000°/se2000°/s. A comunicação através do padrão I2C ocorre a uma taxa máxima de 400 kHz e opera uma faixa de tensão de alimentação de 2,37 V a 3,46 V.

Figura 2 – Protótipo Final do Hardware.



Fonte: Os autores.

Na Tabela 1 é apresentada a visão geral dos componentes utilizados no projeto, com os preços a partir de novembro de 2023.

Tabela 1 – Visão Geral dos Componentes.

Componente	Nome	Preço
Microcontrolador	ESP8266	R\$ 47,40
Carregador	TP4056	R\$ 5,40
Bateria 3.7V, 500mAh	JH752540P	R\$ 39,99
Módulo Step UP	XL6009	R\$ 17,64
I ² c Multiplexador	TCA9548A	R\$ 20,98
IMU x 5	MPU6050	R\$ 67,50
Luva	-	R\$ 1,99
Total		R\$ 213,83

Fonte: Os autores.

Os dados capturados pelos acelerômetros são então processados e utilizados como entrada para algoritmos de classificação de movimentos. Esses algoritmos analisam os padrões de movimento dos dedos e os comparam com padrões conhecidos, permitindo identificar e classificar a ação desejada pelo usuário.

3.2 Comunicação entre os Sistemas e Transmissão dos Dados

A transmissão de dados dos sensores até o sistema de IA, utilizando o protocolo MQTT em conjunto com o NodeMCU ESP8266, ocorre por meio de uma sequência de etapas estruturadas que garantem a comunicação eficiente entre os dispositivos e o processamento dos dados pelo modelo de inteligência artificial.

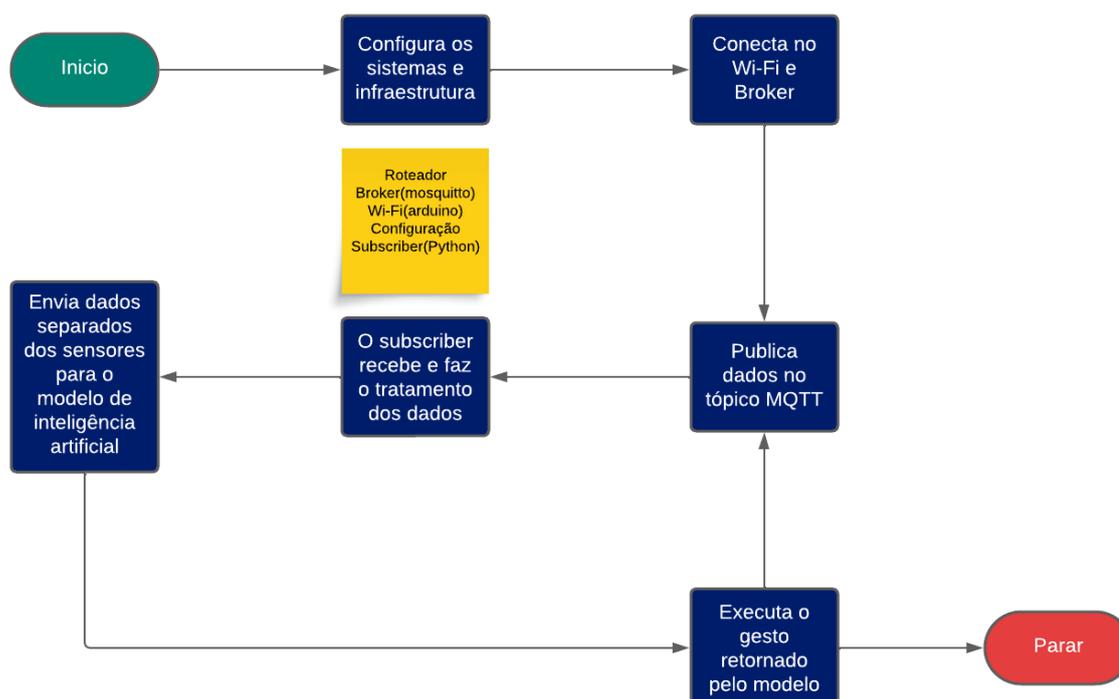
Inicialmente, o ESP8266 precisa se conectar à mesma rede Wi-Fi que o *broker* MQTT, o intermediário responsável por gerenciar o envio e recebimento das mensagens entre os dispositivos. Uma vez estabelecida essa conexão com a rede, o ESP8266 atua como um *publisher*, ou seja, um dispositivo que coleta e publica dados em um tópico específico no *broker*. Esse processo de conexão é crucial para permitir a transmissão de informações dos sensores de forma organizada e em tempo real.

Após essa configuração, o ESP8266 inicia a coleta dos dados dos sensores acelerômetros, que capturam variações de movimento e orientação. Esses dados são organizados em um formato JSON (*JavaScript Object Notation*) e então publicados no tópico MQTT previamente configurado, enviando as informações para o *broker*, que por sua vez, distribui os dados para todos os dispositivos que estejam assinados como *subscribers* do tópico. Nesse cenário, o código Python, que executa em um servidor ou máquina local, age como o *subscriber*. Ao receber a mensagem publicada no tópico, o código Python processa o JSON e, em seguida, envia os dados para o modelo de IA baseado em *transformers*, que já foi previamente carregado no sistema. Esse modelo foi treinado para reconhecer padrões nos dados dos sensores que correspondem a gestos específicos.

Conforme ilustrado na Figura 3, o fluxo de comunicação entre o NodeMCU ESP8266 e o sistema de IA via protocolo MQTT garante a transmissão eficiente dos dados dos sensores até o reconhecimento de gestos. Cada vez que o Python recebe uma nova mensagem do tópico MQTT, os dados dos sensores são processados pelo modelo de IA, que realiza a inferência em tempo real. O modelo avalia se os padrões de movimento correspondem a um gesto previamente aprendido. Se o gesto for identificado, o sistema executa a ação correspondente, como passar ou voltar slides.

Os testes preliminares desempenharam um papel essencial na validação do funcionamento dos componentes e do fluxo de dados do sistema de reconhecimento de gestos. Inicialmente, foi realizada uma verificação individual dos principais componentes do hardware. O Arduino foi testado em conjunto com os acelerômetros, multiplexador, bateria, interruptor e o carregador da bateria, garantindo que cada um operasse dentro dos parâmetros esperados. Um dos pontos críticos foi a verificação do módulo Wi-Fi da placa do Arduino, o qual foi necessário para permitir a comunicação sem fio via protocolo MQTT.

Figura 3 – Fluxograma do Sistema.



Fonte: Os autores.

Para assegurar a integridade da transmissão de dados, foi testado o envio das informações coletadas pelos acelerômetros via cabo USB do Arduino para o computador. Posteriormente, experimentou-se o uso do protocolo MQTT com o *broker* Mosquitto para viabilizar a comunicação entre o Arduino e o algoritmo em Python. A transmissão de dados foi realizada, e os dados foram devidamente recebidos pelo algoritmo, sendo armazenados em formato JSON para facilitar a organização e posterior análise.

3.3 Aprendizado de Máquina (*Machine Learning*)

Este projeto de reconhecimento de gestos manuais utiliza uma arquitetura baseada em *transformers*, empregando PyTorch como *framework* principal. O objetivo é classificar gestos capturados por sensores de aceleração e giroscópio em tempo real, atribuindo a cada gesto uma ação específica, como "passar slide" ou "voltar slide". Para isso, o sistema processa sequências de dados dos sensores, que são então utilizadas para treinar um modelo de aprendizado de máquina capaz de reconhecer padrões nos movimentos.

Os dados de treinamento são organizados em arquivos JSON, e cada arquivo representa um tipo de gesto. Cada item no arquivo corresponde a uma leitura dos sensores, contendo seis valores (*accX*, *accY*, *accZ*, *gyrX*, *gyrY*, *gyrZ*), representando as acelerações e rotações nos eixos X, Y e Z, respectivamente. O sistema carrega esses dados, organiza-os como sequências temporais e associa cada sequência ao gesto correspondente. Esse processo é fundamental para que o modelo possa aprender as diferentes características que distinguem os gestos.

O modelo de reconhecimento de gestos é construído utilizando a arquitetura *Transformer*, especificamente a parte do encoder, que é ideal para processar sequências de dados, capturando dependências temporais e padrões complexos ao longo do tempo. A arquitetura do modelo inclui uma camada de *embedding* que transforma os dados dos sensores, originalmente em seis dimensões, em uma representação de maior dimensionalidade. Em seguida, os dados são processados por várias camadas de um *Transformer Encoder*, com múltiplas cabeças de atenção, que permitem ao modelo focar em diferentes partes da sequência para melhor entender o padrão dos movimentos. Após o processamento, uma camada final realiza a classificação do gesto.

O processo de desenvolvimento envolve etapas fundamentais para a construção de um sistema robusto de aprendizado de máquina. Inicialmente, os dados são carregados e divididos em conjuntos de treinamento e teste, assegurando que o modelo seja avaliado em dados que ele ainda não viu durante o treinamento. Um *DataLoader* personalizado é utilizado para organizar os dados e prepará-los para serem passados em *batches* ao modelo, o que permite um treinamento mais eficiente e controlado.

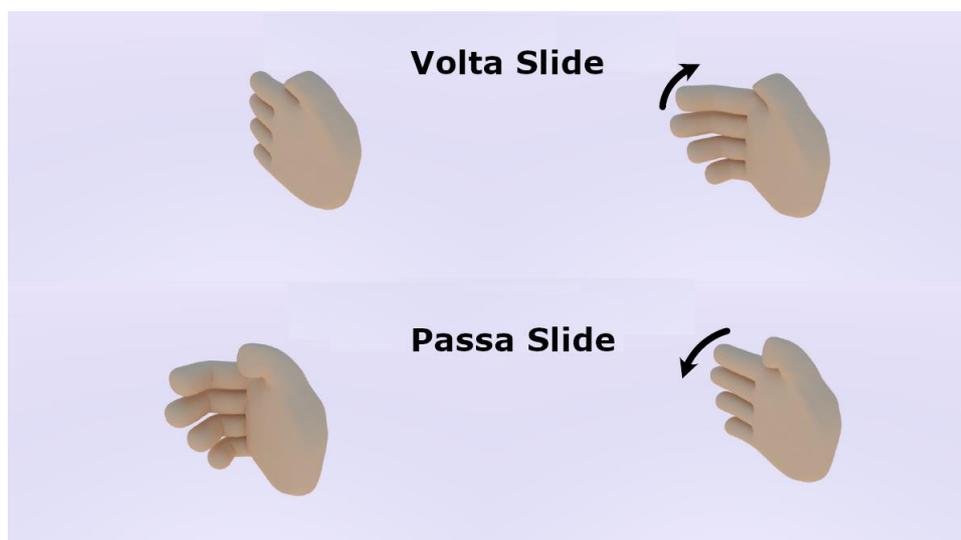
Durante o treinamento, o modelo é otimizado usando o algoritmo de *backpropagation*, no qual a função de perda, *CrossEntropyLoss*, calcula o erro entre as previsões do modelo e os rótulos corretos dos gestos. O otimizador ainda ajusta os pesos do modelo para minimizar esse erro ao longo de várias iterações. Ao final de cada época de treinamento, o erro médio é monitorado para garantir que o modelo está aprendendo corretamente.

Após o treinamento, o modelo é avaliado no conjunto de teste, e a acurácia é calculada para verificar a sua capacidade de generalizar para novos dados. O modelo treinado é então salvo em um arquivo *.pt*, permitindo sua reutilização em aplicações futuras, como o reconhecimento de gestos em tempo real.

3.4 Sistema de Comandos

No sistema de comandos para controle de slides, a principal função envolve a interpretação de gestos específicos para acionar as operações "passar slide" e "voltar slide", como apresentado na Figura 4. Esses comandos são interpretados por meio de sensores, como acelerômetros, que captam o movimento da mão ou dos dedos. A Inteligência Artificial processa esses dados e reconhece padrões, convertendo-os em ações no *software* de apresentação. Assim, ao realizar um gesto, como um movimento da mão para a direita, o sistema passa para o próximo slide, enquanto um movimento contrário retorna ao slide anterior.

A escolha de utilizar apenas dois gestos se deve ao fato de serem os fundamentais para uma apresentação de slides, mas também ao concentrar em um conjunto de gestos essenciais, consegue-se garantir uma maior precisão no reconhecimento e reduzir a complexidade do sistema, sem comprometer sua funcionalidade básica.

Figura 4 – Gestos Mapeados.

Fonte: KRÁLIK (2020, p. 89), adaptada pelos autores.

4 Resultados da solução Proposta

Ao final do treinamento, o modelo foi avaliado no conjunto de testes para verificar sua capacidade de generalizar em novos dados. O modelo alcançou uma acurácia de 99,68%, o que indica uma alta taxa de acerto na classificação dos gestos, conforme os resultados dos testes apresentados na Tabela 2.

Os testes realizados variaram em termos de número de camadas, cabeças de atenção e passagens (*epochs*), permitindo uma análise abrangente do desempenho do modelo sob diferentes condições. A acurácia foi medida em três execuções para cada configuração de teste, com a média sendo calculada para representar a precisão global. O tamanho do conjunto de testes foi de 20% e 30% dos dados disponíveis.

Os melhores resultados foram observados nos testes com menor número de camadas (3), menor número de cabeças de atenção (4) e com um maior número de passagens (50 ou 100), em que a acurácia média ultrapassou 99%, mostrando a eficácia do modelo em classificar corretamente os gestos. Em contrapartida, o aumento no número de camadas para 5 resultou em uma queda significativa de performance em alguns casos, indicando a complexidade do modelo como fator influente na precisão.

O modelo foi treinado utilizando um conjunto de dados dividido em dois grupos: treino (80%), teste (20%) e treino (70%), teste (30%).

Tabela 2 – Testes de treinamento do modelo

Cabeças de Atenção	Camadas	Passagens	Teste	Treino 1	Treino 2	Treino 3	Média
4	3	5	30%	0,98849	0,98849	0,98733	0,98810
4	3	10	30%	0,98676	0,99194	0,97121	0,98330
4	3	30	30%	0,99424	0,99367	0,99079	0,99290
4	3	50	30%	0,99309	0,99482	0,99136	0,99309
4	3	100	30%	0,98733	0,99712	0,99252	0,99232
4	3	5	20%	0,96891	0,98705	0,99136	0,98244
4	3	10	20%	0,98446	0,99396	0,99309	0,99050
4	3	30	20%	0,99223	0,99568	0,99223	0,99338
4	3	50	20%	0,99827	0,99050	0,98705	0,99194
4	3	100	20%	0,99741	0,99655	0,99655	0,99683
4	5	5	30%	0,97409	0,97409	0,97640	0,97486
4	5	10	30%	0,49280	0,70121	0,50720	0,56707
4	5	30	30%	0,50720	0,50835	0,72078	0,57878
4	5	50	30%	0,54059	0,50720	0,50720	0,51833
4	5	100	30%	0,50720			0,50720
4	5	5	20%	0,56995	0,50432	0,55440	0,54289
4	5	10	20%	0,78238	0,50432	0,74698	0,67789
4	5	30	20%	0,75648	0,50604	0,52936	0,59729
4	5	50	20%	0,50432	0,79102	0,61226	0,63587
4	5	100	20%	0,78411	0,78411	0,72712	0,76511
8	3	5	30%	0,98503	0,99079	0,98849	0,98810
8	3	10	30%	0,99482	0,98733	0,98906	0,99040
8	3	30	30%	0,99252	0,99482	0,99252	0,99328
8	3	50	30%	0,99482	0,98791	0,99539	0,99271
8	3	100	30%	0,99136	0,99252	0,99482	0,99290
8	3	5	20%	0,98877	0,99223	0,98791	0,98964
8	3	10	20%	0,99568	0,99050	0,97755	0,98791

8	3	30	20%	0,99482	0,99050	0,99136	0,99223
8	3	50	20%	0,99827	0,99309	0,99827	0,99655
8	5	5	30%	0,60391	0,50720	0,88371	0,66494

Fonte: Os autores.

No sistema de reconhecimento de gestos manuais proposto, os dados de entrada são provenientes de sensores de acelerômetros e giroscópios presentes nos dedos da luva. Esses sensores capturam informações sobre a aceleração e rotação em três eixos (X, Y e Z), totalizando seis dimensões por sensor (accX, accY, accZ, gyrX, gyrY, gyrZ). Como foram utilizados cinco sensores IMU MPU6050, um em cada dedo da mão, a dimensão total dos dados de entrada é composta por 30 parâmetros (6 valores por sensor x 5 sensores). Esses dados são capturados em tempo real e organizados em sequências temporais, que são usadas como *input* para o modelo de *Transformers*, responsável por classificar os gestos.

O número de classes refere-se aos gestos que serão reconhecidos pelo sistema. Como o foco deste projeto é o controle de apresentações de slides, foram definidos dois gestos principais: "passar slide" e "voltar slide". Esses gestos foram escolhidos por serem os fundamentais para essa aplicação, garantindo simplicidade e eficiência. Além disso, limitar o número de classes a dois gestos essenciais facilita o processo de treinamento e aumenta a precisão do reconhecimento, ao focar em um conjunto menor e mais controlado de padrões gestuais.

5 Conclusão

Este projeto se propôs a desenvolver uma interface baseada em gestos das mãos, focado no controle de apresentações de slides. O protótipo foi composto por uma luva equipada com sensores de acelerômetro e giroscópio, integrados a um microcontrolador ESP8266, que transmitiu os dados via protocolo MQTT para um sistema de Inteligência Artificial baseado na arquitetura *Transformer*. A escolha por essa combinação tecnológica foi fundamental para garantir a captura precisa dos movimentos das mãos e o processamento eficiente dos dados em tempo real. Os testes realizados indicaram que o sistema é capaz de identificar gestos com alta precisão, mostrando um potencial significativo para substituir métodos de controle convencionais, como teclado e mouse, por uma interface mais intuitiva e natural.

A análise dos resultados confirmou a eficiência do sistema proposto, que foi capaz de classificar os gestos de forma confiável e em tempo real, oferecendo benefícios práticos para apresentações de slides. A integração da Inteligência Artificial permitiu maior adaptabilidade e precisão no reconhecimento dos gestos, destacando a importância dessa tecnologia em interfaces interativas. O uso de um modelo baseado em *transformers* também se mostrou adequado para lidar com sequências de dados temporais, melhorando a eficiência do reconhecimento.

No entanto, algumas limitações foram identificadas. O modelo de IA ainda pode ser aprimorado para lidar com gestos mais complexos ou variações sutis nos movimentos, o que ampliaria a robustez do sistema. Além disso, a dependência atual da conectividade sem fio via Wi-Fi para a transmissão dos dados poderia ser

substituída por uma arquitetura mais flexível, ampliando sua aplicabilidade em ambientes onde o Wi-Fi não está disponível. Trabalhos futuros poderiam explorar essas melhorias, além de expandir o sistema para outros dispositivos e incorporar sensores adicionais que aumentem a precisão do reconhecimento gestual.

Ao concluir este projeto, ressalta-se que o desenvolvimento de interfaces mais naturais e acessíveis, como o controle por gestos, têm o potencial de transformar a maneira como interagimos com dispositivos tecnológicos. O sistema proposto é um exemplo claro de como essas tecnologias podem modernizar e facilitar o uso de recursos digitais, oferecendo uma experiência mais fluida e intuitiva. Acredita-se que sistemas como este podem se tornar uma ferramenta fundamental em diversas aplicações, especialmente em áreas que demandam interação sem o uso de controles físicos tradicionais ou de realidade aumentada.

Referências

AERO. O que é acelerômetro? Aero, 2023. Disponível em:

<https://aeroengenharia.com/glossario/o-que-e-acelerometro/>. Acesso em: 25 jun. 2024.

BANZI, M; SHILOH, M. Getting Started with Arduino. 3. ed. Gravenstein Highway North, EUA: Maker Media, Inc, 2014.

GOODFELLOW, I; BENGIO, Y; COURVILLE, A. Deep Learning. MIT Press. Estados Unidos: Cambridge, Massachusetts, 2016.

GUIMARÃES, Fábio. Acelerômetro MPU6050 com o Arduino. Mundo Projetado, 2018. Disponível em: <https://mundoprojetado.com.br/acelerometro-mpu6050-arduino/>. Acesso em: 03 mar. 2024.

HUGHES, J. M. Arduino: A Technical Reference. 1. ed. Sebastopol: O'Reilly Media, 2016. Disponível em: <https://www.oreilly.com/library/view/arduino-a-technical/9781491934319/>. Acesso em: 20 set. 2024.

HSU, Tai-Ran. MEMS and Microsystems: design, manufacture, and nanoscale engineering. 2. ed. Hoboken, New Jersey: John Wiley & Sons, Inc., 2008.

KRÁLIK, Matej. Multi-sensor accelerometer-based gesture recognition. Tese de Mestrado, Comenius University, Bratislava, Eslováquia. 2020.

LECUN, Y; BENGIO, Y; HINTON, G. Deep learning. Nature 521, p. 436–444, 2015. Disponível em: <https://www.nature.com/articles/nature14539>. Acesso em: 25 jun. 2024.

MAKIYAMA, Marcio. O que é Arduino?. Victor Vision, 2022. Disponível em: <https://victorvision.com.br/blog/o-que-e-arduino/>. Acesso em: 7 set. 2024.

MEDEIROS, Anna Carolina Soares. Processo de desenvolvimento de gestos para interfaces de usuário. Dissertação (Mestrado em Informática) - Universidade Federal da Paraíba, João Pessoa, 2015.

MITRA, S.; ACHARYA, T. Gesture Recognition: A Survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 37, no. 3, pp. 311-324, May 2007.

QUIRINO, Felipe A; GIRARDI, Alessandro. Luva de captação de gestos para interação humano-computador. In: 71ª Reunião Anual da SBPC - 21 a 27 de julho de 2019, Universidade Federal de Mato Grosso do Sul (UFMS), Campo Grande, MS. 2019.

VASWANI, A. et al. Attention is all you need. In Advances in Neural Information Processing Systems. pp. 5998-6008. 2017.

VIMALI, J.S.; SRINIVASULU, S.; JABEZ, J.; GOWRI, S. Hand Gesture Recognition Control for Computers Using Arduino. In: Jeena Jacob, I., Kolandapalayam Shanmugam, S., Piramuthu, S., Falkowski-Gilski, P. (eds) Data Intelligence and Cognitive Informatics. Algorithms for Intelligent Systems. Springer, Singapore. https://doi.org/10.1007/978-981-15-8530-2_45. 2021.

WEICHERT, Frank; BACHMANN, Daniel; RUDAK, Bartholomäus; FISSELER, Denis. Analysis of the Accuracy and Robustness of the Leap Motion Controller. Sensors 13, no. 5: 6380-6393. <https://doi.org/10.3390/s130506380>. 2013.

ZALEVSKY, Z. et al. Method and System for Object Reconstruction. US20100177164A1, 19 abril 2007.