

SOFTWARE DE ACESSIBILIDADE POR RECONHECIMENTO VISUAL PARA PESSOAS COM DEFICIÊNCIA

Visual Recognition Accessibility Software for People With Disabilities

Lucas Augustini Yamamoto¹, Jonathan Rafael dos Santos², Wellington L. C. Garcia³

¹Lucas Augustini Yamamoto, lucasay2003@gmail.com

²Jonathan Rafael dos Santos, johathanx@outlook.com

³Wellington L. C. Garcia, wellington.unesp@gmail.com

RESUMO

Este trabalho apresenta o desenvolvimento de um sistema de controle de mouse por meio de reconhecimento facial, voltado para promover acessibilidade e inclusão digital para pessoas com deficiência física ou motora. O projeto foi implementado em Python, utilizando bibliotecas de código aberto como OpenCV, MediaPipe e PyAutoGUI, que fornecem recursos para a captura de vídeo e processamento de imagem. A captura das imagens faciais é realizada via webcam, e, com o uso do modelo FaceMesh da biblioteca MediaPipe, o sistema identifica landmarks faciais específicos. Com base na localização dos olhos, o software calcula a posição da íris para mover o cursor e realiza cliques ao detectar o fechamento de um dos olhos. Esta solução é de baixo custo e fácil implementação, oferecendo uma alternativa acessível para que pessoas com deficiência possam interagir de maneira independente com dispositivos digitais.

Palavras-chave: tecnologia assistiva, reconhecimento facial, controle de mouse, acessibilidade, inclusão digital, interação humano-computador.

ABSTRACT

This paper presents the development of a mouse control system using facial recognition, aimed at promoting accessibility and digital inclusion for individuals with physical or motor disabilities. The project was implemented in Python, utilizing open-source libraries such as OpenCV, MediaPipe, and PyAutoGUI, which provide resources for video capture and image processing. Facial images are captured via webcam, and with the use of MediaPipe's FaceMesh model, the system identifies specific facial landmarks. Based on eye position, the software calculates the iris location to move the cursor and performs clicks upon detecting the closure of one eye. This solution is low-cost and easy to implement, offering an accessible alternative for people with disabilities to interact independently with digital devices.

Keywords: assistive technology, facial recognition, mouse control, accessibility, digital inclusion, human-computer interaction.

1 INTRODUÇÃO

Nos anos recentes, houve avanços significativos na área de tecnologia assistiva, especialmente na elaboração de programas focados em promover a acessibilidade e a inclusão de pessoas com deficiências. Nesse contexto, um campo que tem ganhado destaque é o uso de tecnologias de reconhecimento visual, que permitem que as pessoas com deficiência motora ou física possam interagir de forma mais eficiente com dispositivos e ambientes digitais.

Dentre essas tecnologias, os programas de acessibilidade que utilizam reconhecimento visual têm se sobressaído como um recurso eficaz para aprimorar a autonomia e o bem-estar dessas pessoas. Esses programas, que empregam câmeras e algoritmos sofisticados de processamento de imagens, conseguem interpretar o ambiente ao redor do usuário e oferecer informações relevantes em formatos compreensíveis, possibilitando a interação com aparelhos e objetos do dia a dia.

Neste projeto, foi realizado o desenvolvimento de um software desta tecnologia, explorando os fundamentos técnicos, as características e as vantagens que ela proporciona. Adicionalmente, abordamos os principais obstáculos que se apresentaram durante o processo e as possibilidades para aprimoramentos. O objetivo não é apenas mostrar o que já foi realizado, mas também estimular novas investigações e ações que contribuam para tornar o ambiente digital mais acessível e inclusivo para todos.

2 REFERENCIAL TEÓRICO

Esta seção tem como objetivo apresentar os conceitos essenciais que fundamentam a metodologia adotada neste trabalho, além de introduzir as bibliotecas e ferramentas utilizadas no desenvolvimento do sistema. Ao contextualizar as escolhas metodológicas e técnicas, busca-se proporcionar uma compreensão clara do processo de implementação. A descrição das bibliotecas incluirá suas funções principais, justificativas para sua utilização e como elas contribuem para a solução proposta, visando estabelecer uma base sólida para o entendimento e replicação do trabalho por outros desenvolvedores ou pesquisadores.

2.1 Processamento de Imagem e Visão Computacional

O Processamento de Imagem e Visão Computacional são áreas interdisciplinares da ciência da computação e engenharia que se concentram no desenvolvimento de algoritmos, técnicas e sistemas para a análise e interpretação de imagens digitais. Essas áreas têm sido objeto de intenso estudo e aplicação devido ao seu amplo espectro de aplicações em diversos setores, incluindo medicina, automação industrial, segurança, entretenimento e assistência tecnológica.

O Processamento de Imagem, conforme explica Rafael e Richard (2003), refere-se ao conjunto de técnicas utilizadas para realizar operações em imagens digitais, como filtragem, segmentação, restauração e compressão. Isso envolve a manipulação de pixels e características das imagens para melhorar a qualidade visual, extrair informações relevantes e preparar os dados para análise posterior.

De acordo com Richard e Andrew(2004), a Visão Computacional se concentra na interpretação e compreensão do conteúdo visual das imagens. Isso envolve a extração de características, reconhecimento de padrões, detecção de objetos e reconstrução de cena tridimensional a partir de imagens bidimensionais. A visão computacional busca replicar a capacidade humana de interpretar e compreender o mundo visual ao nosso redor, utilizando algoritmos e técnicas computacionais.

Ambas as áreas têm se beneficiado significativamente dos avanços recentes em inteligência artificial e aprendizado de máquina, especialmente com o surgimento de técnicas como redes neurais convolucionais (CNNs), que têm demonstrado desempenho excepcional em tarefas de reconhecimento de padrões e classificação de imagens.

Em resumo, o Processamento de Imagem e Visão Computacional são áreas essenciais para o desenvolvimento de softwares de acessibilidade por reconhecimento visual, oferecendo ferramentas e técnicas poderosas para a interpretação e compreensão do conteúdo visual e facilitando assim a interação de pessoas com deficiência com o mundo digital e físico ao seu redor.

2.2 Interação Humano-Computador (IHC)

A Interação Humano-Computador (IHC) é uma área interdisciplinar que explora a interação entre humanos e computadores. Segundo Elisa, Marcelo, Raquel (2013), a meta principal é conceber e criar interfaces que simplifiquem essa interação de

maneira eficiente e intuitiva. Ao combinar conhecimentos de áreas como design de interfaces, psicologia cognitiva e ergonomia, a IHC busca criar sistemas que atendam às necessidades e expectativas dos usuários.

Como mencionado por Simone e Bruno (2010), a IHC abrange várias áreas, incluindo design centrado no usuário, usabilidade, acessibilidade e interação natural. Ela se concentra em entender as capacidades e limitações dos usuários, bem como os contextos em que a interação ocorre, para projetar interfaces que sejam fáceis de usar e compreender. Isso envolve técnicas como testes de usabilidade, análise de tarefas e prototipagem iterativa.

2.3 Tecnologia Assistiva

A Tecnologia Assistiva é um campo interdisciplinar dedicado a melhorar a qualidade de vida, autonomia e inclusão social de pessoas com deficiência por meio do desenvolvimento e aplicação de soluções tecnológicas. Jean e Giuliano (2020) destacam que essas tecnologias são essenciais para superar barreiras físicas, sensoriais, cognitivas e comunicacionais, permitindo que indivíduos participem de forma mais plena na vida cotidiana. Segundo Frazão et al. (2020), o escopo do campo varia desde dispositivos simples, como bengalas e lupas, até tecnologias avançadas, como próteses robóticas, leitores de tela e sistemas de navegação assistida, sempre com o propósito de promover a inclusão e a independência.

Capovilla (1997) complementa essa visão ao enfatizar que a pesquisa e o desenvolvimento de novos recursos tecnológicos são cruciais para a educação especial. Ele aponta que esses recursos não devem apenas compensar limitações, mas também potencializar as capacidades dos usuários, oferecendo ferramentas que ampliem a interação, o aprendizado e a autonomia. Para pessoas com deficiência visual, por exemplo, isso inclui o uso de softwares educativos, interfaces auditivas e leitores táteis que facilitam a navegação e a interação em diferentes contextos.

Capovilla também sublinha a necessidade de soluções intuitivas e acessíveis, projetadas para atender às demandas específicas dos usuários. Tecnologias Assistivas eficazes devem ser adaptáveis, garantindo que indivíduos com diversas necessidades possam utilizá-las de forma plena. Essa abordagem reforça o papel desses sistemas como instrumentos de inclusão e empoderamento, permitindo que os usuários desenvolvam suas potencialidades e participem ativamente na sociedade.

A Teoria da Tecnologia Assistiva, nesse contexto, fornece um referencial conceitual para guiar o design e a implementação de soluções tecnológicas. Ao unir os princípios apresentados por Jean, Giuliano, Frazão e Capovilla, é possível desenvolver ferramentas que transcendem o caráter funcional e se tornam fundamentais para a inclusão. Essa perspectiva reforça a importância de um design centrado no usuário, com foco na criação de uma sociedade mais inclusiva e equitativa.

2.4 Acessibilidade

De acordo com Garcia *et al* (2018), a acessibilidade é um conjunto de princípios e conceitos fundamentais que visam garantir que todos, independentemente de suas capacidades físicas, sensoriais, cognitivas ou emocionais, tenham igualdade de acesso a oportunidades, recursos e ambientes. Ela reconhece que as barreiras à participação plena não se limitam apenas a impedimentos físicos, mas também incluem barreiras de comunicação, informacionais e atitudinais.

Em seu cerne, como explicado por Lucas e Roberto (2016), a Acessibilidade promove o Desenho Universal, que busca criar ambientes, produtos e serviços que sejam utilizáveis por todas as pessoas, sem a necessidade de adaptações posteriores. Isso implica pensar de forma proativa em como tornar os espaços físicos, tecnológicos e sociais acessíveis desde o momento do planejamento e do design.

Em última análise, a acessibilidade visa empoderar pessoas com deficiência, fornecendo-lhes os recursos e suportes necessários para que possam exercer sua autonomia, tomar decisões e alcançar seus objetivos pessoais e profissionais, contribuindo assim para uma sociedade mais inclusiva e diversificada.

2.5 Tecnologias

O desenvolvimento do projeto baseou-se nos conceitos apresentados na videoaula de Programming Hero (2022), que demonstra a construção de um sistema de controle do mouse utilizando o movimento dos olhos e a integração de bibliotecas como OpenCV e PyAutoGUI. Essa abordagem serviu como referência prática para a implementação do sistema proposto neste trabalho.

2.5.1 Python

Python é uma linguagem de programação de alto nível e multiparadigma, conhecida por sua simplicidade e versatilidade. Criada por Guido van Rossum em 1991, é amplamente utilizada em áreas como ciência de dados, inteligência artificial, desenvolvimento web e automação.

Sua sintaxe clara e uma ampla biblioteca padrão tornam Python ideal tanto para iniciantes quanto para desenvolvedores experientes. No contexto deste projeto, foi escolhida por sua capacidade de integrar bibliotecas como OpenCV, MediaPipe e PyAutoGUI de forma eficiente, além de sua compatibilidade com diferentes sistemas operacionais, facilitando o desenvolvimento e a execução de aplicações científicas e tecnológicas.(VAN ROSSUM, 2024)

2.5.2 OpenCV

A OpenCV (Open Source Computer Vision Library) é uma biblioteca de código aberto amplamente utilizada para tarefas de processamento de imagens e visão computacional. Lançada originalmente pela Intel em 1999, ela possui suporte para várias linguagens de programação, incluindo Python, e oferece uma ampla gama de funcionalidades para manipulação de imagens, detecção de objetos, reconhecimento facial, entre outras.

No contexto deste trabalho, a OpenCV é utilizada para capturar frames de vídeo em tempo real a partir da webcam e realizar operações básicas de pré-processamento nas imagens, como redimensionamento ou conversão para escala de cinza. Sua popularidade se deve à sua eficiência e ao suporte a hardware acelerado, o que permite o desenvolvimento de aplicações em tempo real (BRADSKY; KAEBERLEIN, 2008).

2.5.3 MediaPipe

O MediaPipe é uma biblioteca de machine learning desenvolvida pelo Google, voltada para a criação de soluções multimodais e em tempo real, principalmente em visão computacional e processamento de áudio. A biblioteca oferece uma série de modelos pré-treinados e pipelines otimizados para tarefas como detecção de mãos, reconhecimento facial, rastreamento de poses, entre outros.

O modelo FaceMesh, utilizado neste trabalho, é uma das soluções fornecidas pelo MediaPipe e se destaca por identificar até 468 landmarks faciais em tempo real, mapeando pontos-chave como olhos, sobrancelhas, nariz, e lábios. Essa precisão é essencial para aplicações que dependem de interações baseadas em gestos ou expressões faciais. Além disso, o MediaPipe é projetado para funcionar de forma eficiente em dispositivos de diferentes capacidades, desde smartphones até computadores de alto desempenho (GOOGLE, 2024).

2.5.4 PyAutoGUI

A PyAutoGUI é uma biblioteca Python voltada para automação de interfaces gráficas de usuário (GUI). Ela permite a simulação de interações humanas com o computador, como movimentos do mouse, cliques, entrada de texto e pressionamento de teclas. É amplamente utilizada em projetos de automação de tarefas repetitivas e testes de software.

No presente trabalho, a PyAutoGUI é responsável por traduzir as coordenadas obtidas do reconhecimento facial em movimentos do cursor na tela. Essa funcionalidade simplifica a integração do sistema com o ambiente gráfico do computador, eliminando a necessidade de um dispositivo físico, como o mouse, para interagir com a interface. Sua simplicidade de uso e ampla compatibilidade tornam a biblioteca uma escolha eficiente para este tipo de aplicação (SWAROOP, 2018).

3. METODOLOGIA

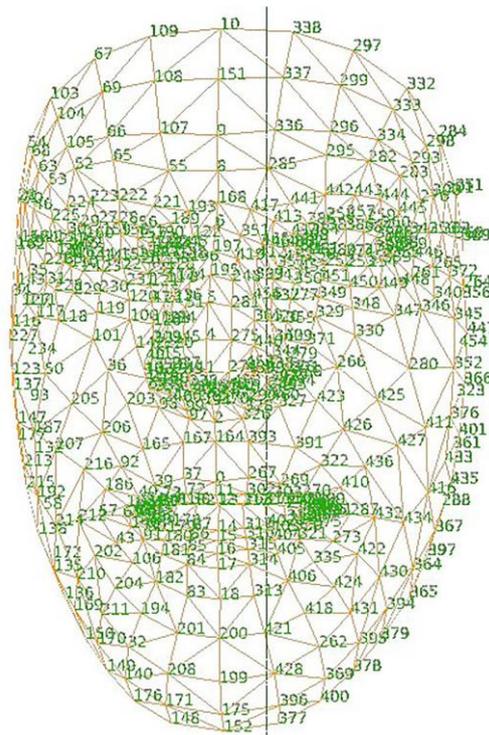
Este trabalho propõe uma alternativa para o controle do cursor do computador, utilizando gestos faciais detectados por meio da webcam. O objetivo é oferecer uma forma mais intuitiva e acessível de interação com o computador, especialmente útil em contextos onde o uso de um mouse convencional não é viável ou desejado.

O projeto foi desenvolvido em Python, versão 3.12.3, uma linguagem amplamente utilizada por sua simplicidade e flexibilidade, sendo ideal para esse tipo de aplicação. Para implementar as funcionalidades necessárias, foram utilizadas as bibliotecas OpenCV, MediaPipe e PyAutoGUI, que são bem estabelecidas na comunidade de desenvolvimento de software. O ambiente de desenvolvimento escolhido foi o PyCharm Community Edition, e a captura das imagens foi feita com a webcam do notebook, o que confere praticidade e portabilidade ao sistema.

A captura de vídeo é realizada com a biblioteca OpenCV, que permite acessar fluxos de vídeo em tempo real de câmeras conectadas ao computador. Isso facilita a aplicação de algoritmos de processamento de imagem imediatamente após a captura, o que é essencial para garantir a fluidez da interação. No contexto deste projeto, a OpenCV trabalha em conjunto com o MediaPipe para realizar o reconhecimento facial.

Após capturar o vídeo, o MediaPipe é utilizado para processar a imagem e identificar pontos de referência no rosto, por meio do modelo FaceMesh. Este modelo já vem pré-treinado e consegue mapear os principais pontos faciais, como os contornos da face e a posição dos olhos, de maneira rápida e precisa. A partir dessas informações, é possível calcular a posição da íris, que será utilizada para controlar o movimento do mouse na tela.

Figura 1 - Mapeamento Facial Realizado pelo MediaPipe



Fonte: https://www.researchgate.net/figure/Media-pipe-face-mesh-solution-map_fig3_377232980

Foram configuradas as *landmarks* de forma em que o olho direito seja detectado para a movimentação do ponteiro, e o clique seja feito quando o olho esquerdo for fechado.

Figura 2 - Landmarks Considerados dos Olhos Direito e Esquerdo



Fonte: Elaborada pelos autores.

Para gerenciar e controlar as funções do sistema operacional, foi utilizada uma biblioteca de automação em Python conhecida como PyAutoGUI. Esta biblioteca fornece uma série de recursos úteis, incluindo a capacidade de mover o cursor do mouse e realizar cliques de forma programada. Isso permite uma maior eficiência na execução de tarefas e na interação com a interface do usuário, contribuindo assim para a melhoria geral do desempenho do sistema.

4. RESULTADOS OBTIDOS

O projeto alcançou o objetivo principal de desenvolver um sistema funcional de controle do mouse por gestos faciais, atendendo às expectativas de acessibilidade e inclusão digital. Durante os testes, foi inicialmente identificado que os usuários precisavam movimentar demais a cabeça para controlar o cursor, o que dificultava a usabilidade do sistema. Para solucionar esse problema, aumentou-se a sensibilidade do sistema, permitindo que movimentos menores fossem suficientes para deslocar o cursor. Após esse ajuste, os resultados melhoraram significativamente, tornando a interação mais confortável e eficiente. No entanto, algumas limitações foram observadas. A distância do usuário em relação à câmera influenciou a precisão do reconhecimento facial, e o uso de óculos impactou a detecção dos olhos em certos casos. Além disso, as bordas da área capturada pela câmera representaram desafios para a navegação do cursor, especialmente em movimentos amplos. Apesar dessas

dificuldades, o sistema demonstrou ser uma solução viável e acessível, com grande potencial para adaptações futuras que possam superar essas limitações e ampliar sua aplicabilidade.

5. CONCLUSÃO

Este projeto de controle de mouse por gestos faciais trouxe uma contribuição para a área de tecnologia assistiva, oferecendo uma solução acessível para pessoas com deficiência. Por meio do reconhecimento visual e de gestos oculares, o sistema permitiu que usuários controlassem o cursor e interagissem apenas com movimentos faciais, sem a necessidade de toques ou controles físicos. Isso representa um passo importante na busca por inclusão digital e maior independência para essas pessoas.

Usando ferramentas de código aberto como OpenCV, MediaPipe e PyAutoGUI, a implementação mostrou que soluções acessíveis e eficazes podem ser construídas com recursos que estão ao alcance da maioria dos desenvolvedores. A simplicidade e a praticidade desse sistema são características que abrem muitas possibilidades, especialmente para quem busca criar experiências mais inclusivas em diversos contextos, como o uso de dispositivos em casa, em ambientes de estudo ou em espaços públicos.

Contudo, foram detectados alguns obstáculos, tais como a exigência de ajustes finos na identificação de gestos, a dependência de condições de iluminação apropriadas e a dificuldade de detecção em usuários que usam óculos. Esses aspectos indicam caminhos para aprimoramentos futuros, como o refinamento dos algoritmos de reconhecimento e a adaptação para diferentes condições de uso.

Em suma, este trabalho reflete o poder da tecnologia quando voltada para o benefício de todos. Mais do que uma solução técnica, ele é um passo em direção a uma sociedade mais inclusiva, onde a tecnologia serve como ponte para a independência e o empoderamento de cada pessoa.

REFERÊNCIAS

GONZALEZ, R. C.; WOODS, R. E. Digital image processing. Uttar Pradesh, India: Pearson, 2018.

HARTLEY, R.; ZISSERMAN, A. Multiple view geometry in computer vision. New York: Cambridge University Press, 2017.

PINHO, R.; TAVARES, J.; CORREIA, M. Introdução à Análise de Movimento usando Visão Computacional. [s.l.] FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO, jun. 2004. Disponível em: <https://web.fe.up.pt/~tavares/downloads/publications/relatorios/Introducao_RPinho_PhD.pdf>. Acesso em: 1 nov. 2024.

ALBERGARIA, E.; BAX, M.; PRATES, R. Interação Humano Computador na Ciência da Informação. 2013.

BARBOSA, S. D. J.; SILVA, B. S. Interação Humano-Computador. Rio de Janeiro: Editora Campus-Elsevier, 2010.

LAPELUCCI, J.; PINTO, G. Tecnologia assistiva e seus conceitos e fundamentos. Revista Interface Tecnológica, v. 17, p. 317-324, 2020. DOI: 10.31510/infa.v17i2.1020.

Frazão, Aline & Zaqueu, Livia & Mendonça, Ísis & Silva, Thays & Morais da Silveira, Francisca. (2020). TECNOLOGIA ASSISTIVA: APLICATIVOS INOVADORES PARA ESTUDANTES COM DEFICIÊNCIA VISUAL / ASSISTIVE TECHNOLOGY: INNOVATIVE APPLICATIONS FOR STUDENTS WITH VISUAL DISABILITIES. Brazilian Journal of Development. 6. 85076-85089. 10.34117/bjdv6n11-066.

GARCIA, R.; BACARIN, A.; LEONARDO, N. Acessibilidade e permanência na educação superior: percepção de estudantes com deficiência. Psicologia Escolar e Educacional, v. 22, p. 33-40, 2018. DOI: 10.1590/2175-3539/2018/035.

FEITOSA, L.; RIGHI, R. Acessibilidade arquitetônica e desenho universal no mundo e Brasil. Revista Nacional de Gerenciamento de Cidades, v. 4, 2016. DOI: 10.17271/2318847242820161371.

BRADSKEY, G.; KAEBERLEIN, A. Learning OpenCV: Computer Vision with the OpenCV Library. 1. ed. O'Reilly Media, 2008.

GOOGLE. MediaPipe Documentation. Disponível em: <https://mediapipe.dev>. Acesso em: 18 nov. 2024.

SWAROOP, C. Automate the Boring Stuff with Python: Practical Programming for Total Beginners. 2. ed. No Starch Press, 2018.

VAN ROSSUM, G.; DRAKE, F. L. Python Tutorial. Python Software Foundation, 2024. Disponível em: <https://docs.python.org/3/tutorial/>. Acesso em: 18 nov. 2024.

PROGRAMMING HERO. BUILD A MOUSE USING YOUR EYE - Python Project. Disponível em: <<https://www.youtube.com/watch?v=k3PcVruvZCs>>. Acesso em: 18 nov. 2024.

FUNDAÇÃO EDUCACIONAL DE FERNANDÓPOLIS
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

CAPOVILLA, Fernando C. Pesquisa e desenvolvimento de novos recursos tecnológicos para educação especial: boas novas para pesquisadores, clínicos, professores, pais e alunos. *Boletim Educação/UNESP*, n. 1, 1997.