

ANÁLISE EXPERIMENTAL DA INTERAÇÃO HUMANO-COMPUTADOR ATRAVÉS DA VISÃO COMPUTACIONAL E RASTREAMENTO DO OLHAR

Jefferson Lorençoni de Moraes¹
Lucas Felipe Oliveira Rolindo²
Rayssa Oliveira Silva³
Gustavo Nunes Muniz⁴

Resumo: O avanço da visão computacional tem se mostrado fundamental para o desenvolvimento tecnológico em diversas áreas, incluindo a interação humano-computador. Este estudo experimental explora a aplicação da visão computacional para rastreamento do olhar em tempo real e sua utilização no controle de dispositivos físicos. O experimento foi conduzido utilizando uma placa Arduino Uno, um motor DC, resistores de 220 ohms e uma ponte H para as movimentações físicas, integrados com programação em Python, utilizando as bibliotecas OpenCV (cv2), NumPy (np) e PySerial. A metodologia detalha a aquisição de imagens da face, a detecção de *landmarks* faciais e a interpretação dos movimentos oculares para acionar o motor DC na direção correspondente. Os resultados obtidos demonstram a viabilidade de uma interação intuitiva e não invasiva baseada no olhar para o controle de um motor DC, abrindo caminho para diversas aplicações futuras em automação, assistência e acessibilidade. Esta pesquisa valida experimentalmente a integração de visão computacional em tempo real com hardware de baixo custo (Arduino) para controle físico, apresentando uma metodologia detalhada e discutindo os desafios e potenciais desta abordagem para a interação humano-computador.

Palavras-chave: Visão Computacional. Rastreamento Ocular. Interação Humano-Computador. Arduino. OpenCV.

EXPERIMENTAL ANALYSIS OF HUMAN-COMPUTER INTERACTION THROUGH COMPUTATIONAL VISION AND EYE TRACKING

Abstract: The advancement of computer vision has proven to be fundamental for technological development in several areas, including human-computer interaction. This experimental study explores the application of computer vision for real-time eye tracking and its use in the control of physical devices. The experiment was conducted using an Arduino Uno board, a DC motor, 220 ohms resistors and an H-bridge for physical movements, integrated with Python programming, using the OpenCV (cv2), NumPy (np) and PySerial libraries. The methodology details the acquisition of facial images, the detection of facial landmarks and the interpretation of eye movements to drive the DC motor in the corresponding direction. The results obtained demonstrate the feasibility of an intuitive and non-invasive interaction based on gaze for the control of a DC motor, paving the way for several future applications in automation, assistance and accessibility. This research experimentally validates the integration of real-time computer vision with low-cost hardware (Arduino) for physical control, presenting a detailed methodology and discussing the challenges and potential of this approach for human-computer interaction.

KEYWORDS: Computer Vision. Eye Tracking. Human-Computer Interaction. Arduino. OpenCV.

¹ Professor Assistente do Centro Universitário de Goiás – UNIGOIÁS. Doutor em Física Quântica Pura e Avançada pela UFG. Mestre em Engenharia Mecânica pela UFU-MG. Especialistas em Engenharias Elétrica, Automação, IA e Machine Learning, Energias Renovável e Gestão de Projetos pela Anhanguera. Lattes: lattes.cnpq.br/9489300453845705. Orcid: <https://orcid.org/0000-0001-9088-5399> E-mail: jefferson.morais@unigoias.com.br

² Discente do curso de Engenharia de Software do Centro Universitário de Goiás – UNIGOIÁS. E-mail: luckfeliquiz@gmail.com; Lattes: <http://lattes.cnpq.br/5030921690569683> Orcid: <https://orcid.org/0009-0002-1926-8253>

³ Discente do 3º ano do ensino médio do Colégio Estadual Professora Maria Luiza dos Santos Silva - CEPMLSS E-mail: r4yolive@gmail.com Lattes: <https://lattes.cnpq.br/1602066839138351> Orcid: <https://orcid.org/0009-0001-9013-9334>

⁴ Discente do curso de Análise e Desenvolvimento de Sistemas do Centro Universitário de Goiás – UNIGOIÁS. E-mail: gustavomuniznuni@gmail.com; Lattes: <https://lattes.cnpq.br/7086375979263336> Orcid: <https://orcid.org/0009-0004-6528-3475>

INTRODUÇÃO

O campo da visão computacional tem experimentado um crescimento exponencial nas últimas décadas, transformando fundamentalmente a forma como interagimos com o mundo digital e físico. A capacidade de máquinas "enxergarem" e interpretar imagens e vídeos abriu portas para uma vasta gama de aplicações, desde sistemas de segurança e veículos autônomos até a medicina e a agricultura. Um dos avanços mais impactantes da visão computacional é o desenvolvimento de sistemas de interação humano-computador mais intuitivos e naturais, especialmente através do rastreamento do olhar (M. Kumar-2021 & J. J. Jijesh-2020).

A robótica, um campo intimamente ligados à visão computacional, tem revolucionado a sociedade contemporânea de maneiras profundas e multifacetadas. O termo "robot" surgiu na peça teatral "R.U.R. (Rossum's Universal Robots)" do escritor checoslovaco Karel Čapek em 1920, inicialmente concebendo robôs como "pessoas artificiais criadas a partir de matéria orgânica" cujo propósito era "proporcionar mão de obra barata" para libertar a humanidade do trabalho. Historicamente, a ideia de robôs estava enraizada na noção de escravos, máquinas otimizadas para serem exploradas e produzirem para o benefício humano. Essa conceptualização inicial, embora problemática, lançou as bases para o desenvolvimento de autômatos que hoje desempenham um papel crucial em diversas esferas da sociedade (M. Kumar-2021 & S. Sood-2021).

Atualmente, os robôs estão presentes em fábricas, hospitais, residências e até em ambientes perigosos e remotos. Eles automatizam processos, aumentam a eficiência e a segurança, e realizam tarefas que seriam impossíveis ou perigosas para humanos. A inteligência artificial e a cibernética continuam a oferecer avanços que podem, em última análise, levar à possibilidade de autômatos racionais e sensíveis, com o potencial de atingir ou até superar a inteligência humana. Essa evolução, no entanto, levanta questões éticas e morais complexas, especialmente sobre a interação entre as próprias inteligências artificiais, um campo que Ashrafian (2014) denomina "AlonAI" (Artificial Intelligence-on-Artificial Intelligence) (A. Rao et al-2021 & S. Sood-2021 & D.A. B. Oliveira-2022)

A relevância da visão computacional reside não apenas na automação de tarefas, mas também na sua capacidade de extrair informações valiosas do comportamento humano. Os movimentos oculares, em particular, oferecem uma rica fonte de dados sobre a atenção, a intenção e o interesse de um indivíduo em relação ao conteúdo visual. Como Yun et al. (2013) afirmam, "o comportamento do usuário durante a visualização natural de imagens contém uma

abundância de informações sobre o conteúdo das imagens, bem como informações relacionadas à intenção do usuário e à importância do conteúdo definida pelo usuário". Essa capacidade de inferir informações a partir do olhar humano tem implicações significativas para o aprimoramento de interfaces e para a criação de sistemas mais responsivos e adaptáveis (A. Rao et al-2021 & D. A. B. Oliveira-2021).

Este artigo apresenta um estudo experimental focado no desenvolvimento de um sistema de rastreamento do olhar em tempo real, que permite controlar um dispositivo físico (um motor DC) com base nos movimentos oculares. O objetivo principal é demonstrar a viabilidade e a eficácia dessa interação não invasiva e explorar suas potencialidades e desafios. A pesquisa detalha a metodologia empregada, incluindo a configuração de hardware, a implementação de software e a análise dos dados. As equações matemáticas das *landmarks* faciais são apresentadas para formalizar o processo de detecção e rastreamento. Ao final, discutiremos os desafios e as contribuições deste estudo para o avanço da interação humano-computador e as futuras colaborações que podem surgir a partir desta abordagem.

MATERIAIS E MÉTODOS

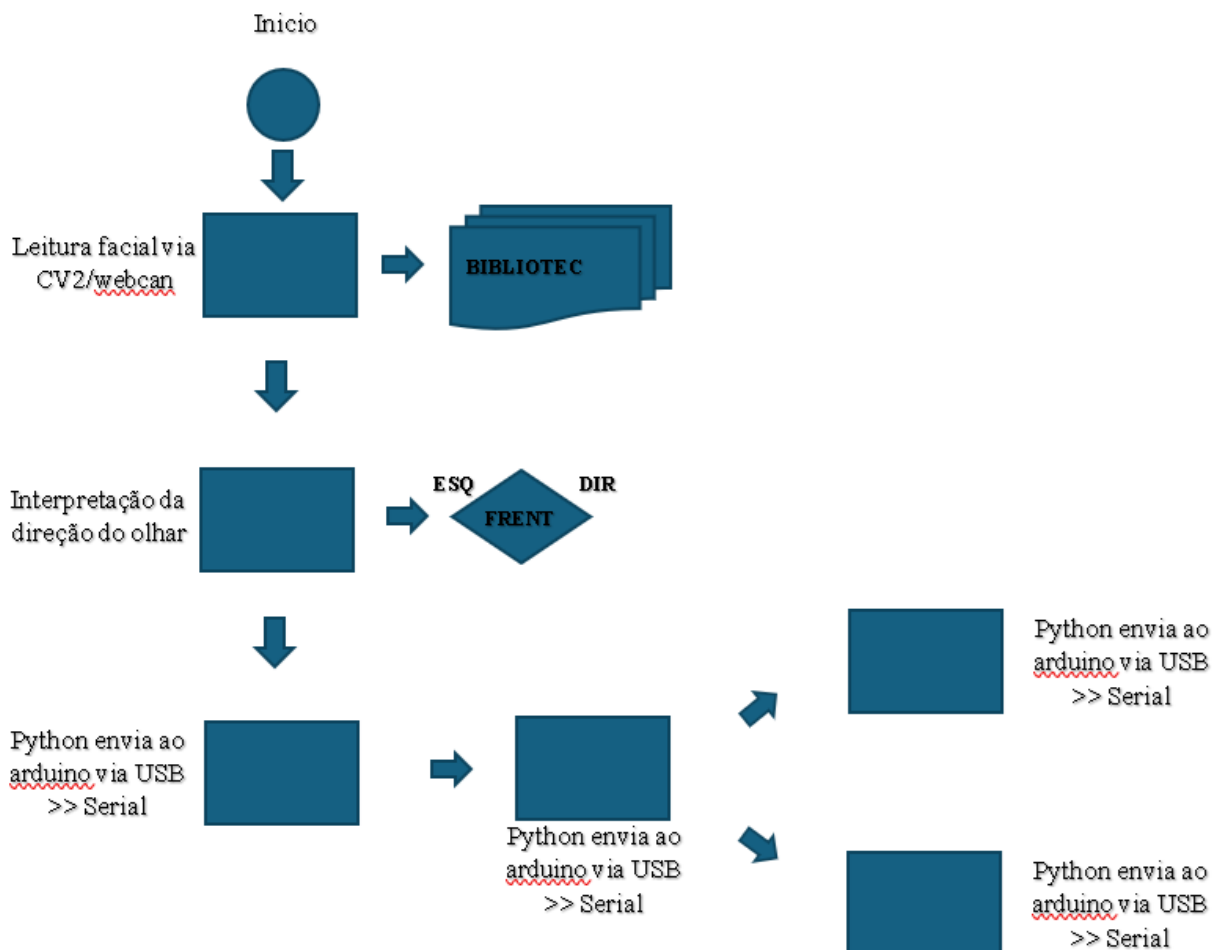
Explicações Gerais:

- **Abordagem da Pesquisa:** Qualitativa e experimental.
- **Tipo de Pesquisa:** Exploratória e descritiva.
- **Técnicas de Pesquisa:** Estudo de caso e experimentação prática.

O sistema desenvolvido neste estudo tem como objetivo rastrear os movimentos dos olhos do usuário em tempo real e traduzi-los em ações de controle para um motor DC. A configuração experimental consistiu em uma combinação de hardware e software, conforme detalhado a seguir.

A arquitetura geral do sistema proposto, ilustrando a interação entre os componentes de hardware e software, é apresentada na figura 01.

Figura 01. Diagrama de Arquitetura do Hardware e Software.



Fonte: Autores (2025).

O fluxo se inicia com a captura de imagem pela webcam, seguida pelo processamento no notebook utilizando Python e OpenCV para detecção facial, identificação de landmarks e interpretação do movimento ocular. Comandos derivados dessa interpretação são enviados via comunicação serial para a placa Arduino Uno, que, por sua vez, controla uma Ponte H (como o L298N) responsável pelo acionamento direcionado do motor DC.

Hardware Utilizado:

- **Notebook:** Um notebook equipado com placa de vídeo AMD Radeon Graphics 5, 8GB de memória RAM e armazenamento SSD PCIe de 256GB. Este hardware foi

selecionado para garantir capacidade de processamento gráfico e computacional adequada para as operações de visão computacional em tempo real.

- **Arduino Uno:** Uma placa de microcontrolador de código aberto, servindo como a interface entre o computador e os componentes físicos. A comunicação entre o Python e o Arduino foi estabelecida via porta serial.
- **Motor DC:** Um motor de corrente contínua, utilizado como atuador físico para demonstrar o controle de movimento direcional.
- **Resistores:** Resistores de 220 Ohm, empregados para limitar a corrente e proteger os componentes eletrônicos.
- **Ponte H:** Um circuito eletrônico (neste caso, provavelmente um chip L298N ou similar) que permite controlar a direção de rotação do motor DC, além de regular a velocidade.

Software e Bibliotecas:

A implementação do software foi realizada em Python, utilizando as seguintes bibliotecas:

- **OpenCV (cv2):** Essencial para as tarefas de visão computacional, incluindo captura de vídeo da webcam, processamento de imagens e detecção de *landmarks* faciais. O OpenCV é amplamente utilizado em aplicações de visão computacional e inteligência artificial.
- **NumPy (np):** Biblioteca fundamental para computação numérica em Python, utilizada para manipulação eficiente de arrays multidimensionais, que são a base das imagens e dos dados de *landmarks*.
- **PySerial:** Biblioteca Python para comunicação serial, utilizada para estabelecer a conexão e enviar comandos do script Python para a placa Arduino Uno.

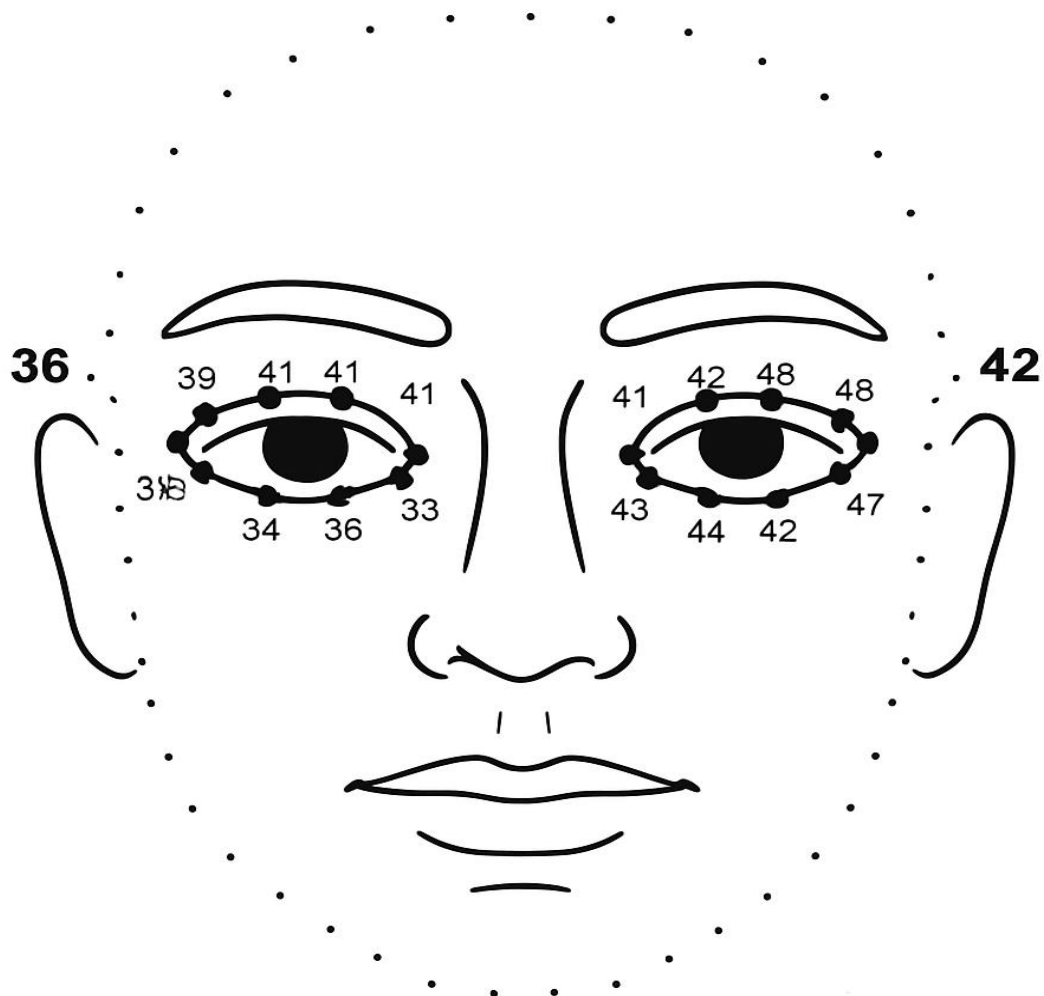
Procedimento Experimental:

O experimento seguiu os seguintes passos principais:

1. **Captura de Imagem:** A câmera do notebook foi utilizada para capturar um *stream* de vídeo da face do usuário em tempo real. O OpenCV foi configurado para acessar a webcam e processar os frames continuamente.
2. **Deteção Facial e de Landmarks:** Para rastrear os movimentos dos olhos, foi empregado um modelo pré-treinado de detecção de faces (e.g., Haar Cascade Frontal Face Classifier ou modelo baseado em Dlib) para identificar a região da face na imagem. Uma vez a face detectada, foram utilizadas técnicas para identificar os *landmarks*

faciais. *Landmarks* são pontos característicos no rosto que são usados para mapear características faciais, como o contorno dos olhos, nariz e boca. Esses pontos são numerados e permitem uma representação geométrica precisa da face. Conforme na figura 02.

Figura 02. Diagrama de Landmarks gerado



Fonte: Autores (2025).

As coordenadas (x, y) dos *landmarks* específicos dos olhos (pontos 36 a 41 para o olho esquerdo e 42 a 47 para o olho direito, segundo o modelo de 68 pontos do Dlib) são particularmente importantes para determinar a direção do olhar. A biblioteca OpenCV, em conjunto com modelos como o `shape_predictor_68_face_landmarks.dat` do Dlib, permite obter essas coordenadas após a detecção da face.

3. **Cálculo e Normalização das Coordenadas:** As coordenadas dos *landmarks* dos olhos são usadas para calcular a posição central de cada olho. Para mitigar o impacto de movimentos da cabeça e variações na distância da câmera, as coordenadas são normalizadas em relação à dimensão da face ou da tela.

A normalização das coordenadas dos *landmarks* é crucial para garantir que o movimento do motor DC seja consistente, independentemente da distância ou posição da cabeça do usuário. As coordenadas (x_i , y_i) de cada *landmark* são obtidas. Para os olhos, podemos focar nas regiões dos olhos. Por exemplo, para calcular a inclinação do olhar, pode-se usar a diferença nas coordenadas y dos pontos mais externos do olho.

A região dos olhos é delimitada pelos *landmarks* correspondentes. Por exemplo, para o olho esquerdo, os pontos 36 a 41 do modelo de 68 *landmarks* são utilizados. Para o olho direito, os pontos 42 a 47.

A inclinação do olhar pode ser determinada calculando a diferença nas coordenadas y entre pontos específicos da borda do olho, como o canto externo e interno, ou o ponto mais alto e mais baixo da pálpebra, e a média dos pontos para determinar a centralização. Por exemplo, para determinar se o olho está se movendo para cima ou para baixo, pode-se analisar a relação entre o centro do olho e o ponto central da pálpebra, normalizando essa distância.

4. **Interpretação dos Movimentos Oculares:** Com base nas coordenadas normalizadas, o sistema interpreta os movimentos dos olhos (esquerda, direita, cima, baixo). Thresholds (limiares) foram empregados para determinar a direção do olhar a partir das variações nas coordenadas. Estes limiares foram definidos empiricamente durante fases de teste e calibração preliminares, buscando um equilíbrio entre sensibilidade e estabilidade na resposta aos movimentos oculares do usuário. Por exemplo, uma variação significativa e sustentada nas coordenadas X para a esquerda, excedendo o limiar estabelecido, seria interpretada como um movimento intencional para a esquerda. Uma análise similar é aplicada para as direções direita, cima e baixo, comparando as variações das coordenadas normalizadas com os respectivos limiares.
5. **Comunicação com o Arduino:** Os comandos de direção (esquerda, direita, cima, baixo, parado) são enviados do script Python para o Arduino via porta serial. O Arduino, por sua vez, recebe esses comandos e aciona a ponte H para controlar o motor DC na direção desejada.

O código Arduino é programado para interpretar os bytes recebidos pela porta serial (e.g., 'L' para esquerda, 'R' para direita) e acionar os pinos da ponte H de acordo, controlando a direção de rotação do motor DC. O PySerial no Python lida com o envio desses caracteres.

6. **Loop Contínuo:** O processo de captura, detecção, interpretação e envio de comandos ocorre em um loop contínuo, proporcionando uma experiência de controle em tempo real.
7. **Considerações sobre Robustez e Limitações:** É importante notar que o desempenho do sistema pode ser influenciado por fatores externos e condições operacionais. Situações como a saída do rosto do usuário do campo de visão da webcam ou falhas temporárias na detecção precisa dos landmarks (devido a oclusões, iluminação inadequada ou movimentos bruscos) podem interromper o fluxo de controle. No presente estudo, o sistema foi configurado para interromper o envio de comandos ao Arduino caso a detecção facial ou de landmarks falhe, retornando ao estado de busca ativa. Falhas na comunicação serial, embora não explicitamente tratadas com mecanismos de recuperação neste protótipo, são minimizadas pelo uso de uma conexão USB estável e taxas de comunicação padrão. Futuros trabalhos poderiam explorar estratégias mais sofisticadas para lidar com essas condições limite, como algoritmos de predição de movimento ou mecanismos de handshake na comunicação serial para garantir a entrega de comandos.

Equações Matemáticas para *Landmarks* Faciais:

A detecção de *landmarks* faciais é um processo fundamental. Embora as bibliotecas como dlib e OpenCV forneçam funções pré-implementadas, a compreensão da matemática subjacente é crucial.

Considerando um conjunto de *landmarks* faciais detectados, representados por suas coordenadas 2D (x_k, y_k), para $k=1, \dots, N$, onde N é o número total de *landmarks*. Para os olhos, podemos isolar os pontos relevantes.

1. **Centro do Olho (exemplo):** Para um olho, se tivermos os pontos (x_1, y_1), ..., (x_M, y_M), o centroide pode ser aproximado por:

$$C_x = \frac{1}{M} \sum_{i=1}^M x_i$$

$$C_y = \frac{1}{M} \sum_{i=1}^M y_i$$

- 2. Relação de Aspecto do Olho (EAR - Eye Aspect Ratio):** Uma métrica comum para detectar se um olho está aberto ou fechado (e, por extensão, o piscar) é a Relação de Aspecto do Olho (EAR), introduzida por Tereza Soukupová e Jan Čech (2016). Ela é calculada com base em seis pontos *landmarks* ao redor de cada olho:

$$EAR = \frac{||P2-P6|| + ||P3-P5||}{2 \cdot ||P1-P4||}$$

Onde $P_1, \dots, P_6$ são os pontos *landmarks* do olho, conforme a numeração específica (por exemplo, 37, 38, 39, 40, 41, 42 para o olho esquerdo no modelo de 68 pontos). P_1 e P_4 representam os cantos horizontais do olho, enquanto P_2, P_3, P_5, P_6 representam os pontos verticais. A distância euclidiana entre os pontos é denotada por $|| \cdot ||$. Um valor baixo de EAR indica que o olho está fechado.

40

3. Detecção de Movimento Horizontal do Olhar:

Para detectar movimentos horizontais, podemos analisar a posição do centroide do olho em relação ao centro da face, ou a proporção da pupila dentro da caixa delimitadora do olho.

Se L_x, L_y e R_x, R_y forem os centros dos olhos esquerdo e direito, respectivamente, e F_x, F_y for o centro da face, podemos calcular a distância relativa.

$$D_{Left} = L_x - F_x$$

$$D_{Right} = R_x - F_x$$

Movimentos para a esquerda ou direita seriam inferidos por variações significativas em D_{Left} e D_{Right} ao longo do tempo.

A implementação pode usar a distância euclidiana entre os pontos de referência para determinar se o olho está se movendo para a esquerda ou para a direita, monitorando a posição relativa dos pontos internos e externos do olho. A detecção do centro da pupila (com algoritmos como *circle Hough transform* ou *pupil-glint tracking*) dentro da região delimitada pelos *landmarks* do olho também pode refinar a detecção da direção do olhar.

Essas equações e métricas fornecem a base para a interpretação dos movimentos oculares, permitindo que o sistema traduza o olhar do usuário em comandos acionáveis para o motor DC.

CONCLUSÃO

Em suma, as técnicas de otimização discreta, como a programação dinâmica e os algoritmos gráficos, decorrem da exploração da estrutura específica de problemas importantes. Esses algoritmos geralmente oferecem fortes garantias sobre a qualidade das soluções encontradas, o que permite uma separação clara entre como um problema é formulado (a escolha da função objetivo) e como o problema é resolvido (a escolha do algoritmo de otimização).

Os métodos são versáteis e surgem em aplicações de visão muito diferentes; por exemplo, a correspondência de sequência por programação dinâmica pode ser usada tanto para reconhecimento de forma quanto para estéreo. Da mesma forma, algoritmos de corte mínimo podem ser usados para estéreo e segmentação de imagem binária. Esses problemas aparentemente não relacionados compartilham uma estrutura que pode ser explorada por métodos de otimização discreta.

O estudo experimental apresentado demonstra a viabilidade e o potencial da interação humano-computador baseada no rastreamento do olhar em tempo real. A integração de hardware como Arduino Uno, motor DC e ponte H com software Python e as bibliotecas OpenCV, NumPy e PySerial, permitiu a criação de um sistema funcional onde o movimento dos olhos do usuário controla a direção de um motor físico. Essa abordagem não invasiva e intuitiva abre um vasto leque de possibilidades para o futuro da tecnologia.

REFERÊNCIAS

- AGARWALA, A. et al. Interactive digital photomontage. *ACM Transactions on Graphics*, v. 23, n. 3, p. 292-300, 2004.
- ALI, S.; EL FATAH, M. A.; KHAIRY, W. A.; SALEM, M. R. Key determinants of client satisfaction with primary healthcare services. *International Journal of Behavioural and Healthcare Research*, v. 6, n. 3/4, p. 237-250, 2018.
- AZAM, I.; KHAN, S. A. Feature extraction trends for intelligent facial expression recognition: A survey. *Informatica*, v. 42, n. 4, 2018. DOI: 10.31449/inf.v42i4.2037.
- BARTNECK, C.; KUMAR, Y.; SER, Q. M.; WOODWARD, G.; SPARROW, R.; WANG, S.; EYSSSEL, F. Robots and racism. In: PROCEEDINGS OF THE 2018 ACM/IEEE

INTERNATIONAL CONFERENCE ON HUMAN-ROBOT INTERACTION (HRI '18), 2018. p. 196-204. Disponível em: <https://doi.org/10.1145/3171221.3171260>.

BERG, A. C. et al. Understanding and predicting importance in images. In: COMPUTER VISION AND PATTERN RECOGNITION (CVPR), 2012. Proceedings... p. 3562-3569, 2012.

CHRISTOFOROU, E. G.; MÜLLER, A. Robot and robotics: the origin and beyond. In: RODIC, A.; BORANGIU, T. (Eds.). Raad 2016: Advances in robot design and intelligent control. Cham: Springer International Publishing, 2017. p. 613-621. Disponível em: https://link-springer-com.proxy.library.uu.nl/chapter/10.1007/978-3-319-49058-8_67.

CHOI, M. J.; LIM, J. J.; TORRALBA, A.; WILLSKY, A. S. Exploiting hierarchical context on a large database of object categories. In: COMPUTER VISION AND PATTERN RECOGNITION (CVPR), 2010. Proceedings... 2010.

DARRELL, T. J.; DEMIRDJIAN, D.; CHECKA, N.; FELZENSZWALB, P. F. Plan-view trajectory estimation with dense stereo background models. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION, 2001. Proceedings... 2001.

IJESH, J. J. et al. Development of machine learning based fruit detection and grading system. In: INTERNATIONAL CONFERENCE ON RECENT TRENDS ON ELECTRONICS, INFORMATION, COMMUNICATION & TECHNOLOGY (RTEICT), 2020. Proceedings... p. 403-407, 2020.

KUMAR, M.; GOYAL, T.; GUPTA, P. Facial emotion recognition in real-time using deep convolution networks. *International Journal of Behavioural and Healthcare Research*, v. 7, n. 3, p. 187-199, 2021.

NIKITHA, M.; ROOPA SRI, S.; UMA MAHESHWARI, B. Fruit recognition and grade of disease detection using Inception V3 model. In: INTERNATIONAL CONFERENCE ON ELECTRONICS COMMUNICATION AND AEROSPACE TECHNOLOGY (ICECA), 2019. Proceedings... v. 11, n. 3, p. 1040-1043, 2019.

OLIVEIRA, D. A. B. et al. A review of deep learning algorithms for computer vision systems in livestock. *Livestock Science*, 2021.

RAO, A.; KULKARNI, S. A hybrid approach for plant leaf disease detection and classification using digital image processing methods. *The International Journal of Electrical Engineering & Education*, p. 0020720920953126, 2020.

RAO, A. et al. Studying the effects of self-attention for medical image analysis. In: ICCVW, 2021. Proceedings... p. 3416-3425, 2021.

SOOD, S.; SINGH, H.; MALARVEL, M. Image quality enhancement for wheat rust diseased images using histogram equalization technique. In: INTERNATIONAL CONFERENCE ON COMPUTING METHODOLOGIES AND COMMUNICATION (ICCMC), 2021. Proceedings... p. 1035-1042, 2021.

SOOD, S.; SINGH, H.; MALARVEL, M. Image quality enhancement for wheat rust diseased images using histogram equalization technique. In: INTERNATIONAL CONFERENCE ON COMPUTING METHODOLOGIES AND COMMUNICATION (ICCMC), 2021. Proceedings... p. 1035-1042, 2021. (Duplicado da referência anterior).

SUNARYO, M.; HARIADI, M. Preprocessing on digital image using histogram equalization: An experiment study on MRI brain image. *International Journal of Computer Science and Information Technologies*, v. 7, n. 4, p. 1723-1727, 2016.

YUN, K. et al. Studying relationships between human gaze, description, and computer vision. In: COMPUTER VISION AND PATTERN RECOGNITION (CVPR), 2013. Proceedings... p. 739-746, 2013.

Recebido: 23 de junho de 2025.

Aceito: 29 de setembro de 2025