



UNIVERSIDADE FEDERAL DO PARÁ
CAMPUS UNIVERSITÁRIO DE CASTANHAL
FACULDADE DE COMPUTAÇÃO
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

ANNE LIVIA DA FONSECA MACEDO

**TÉCNICAS DE VISÃO COMPUTACIONAL APLICADAS NA DETECÇÃO E
RASTREAMENTO OCULAR PARA A INCLUSÃO DIGITAL DE PESSOAS COM
DEFICIÊNCIA MOTORA.**

Castanhal - Pará
2021

ANNE LIVIA DA FONSECA MACEDO

**TÉCNICAS DE VISÃO COMPUTACIONAL APLICADAS NA DETECÇÃO E
RASTREAMENTO OCULAR PARA A INCLUSÃO DIGITAL DE PESSOAS COM
DEFICIÊNCIA MOTORA.**

Trabalho de Conclusão de Curso apresentado à Faculdade de Computação da Universidade Federal do Pará – Campus Castanhal, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Igor Ruiz Gomes

**Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD
Sistema de Bibliotecas da Universidade Federal do Pará
Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a) autor(a)**

M141t Macedo, Anne Livia da Fonseca
Técnicas de Visão Computacional aplicadas na detecção e rastreamento ocular para a inclusão digital de pessoas com deficiência motora / Anne Livia da Fonseca Macedo. — 2021.
71 f. : il. color.

Orientador(a): Prof. Dr. Igor Ruiz Gomes
Trabalho de Conclusão de Curso (Graduação) - Faculdade de Sistemas de Informação, Campus Universitário de Castanhal, Universidade Federal do Pará, Castanhal, 2021.

1. Inclusão Digital. 2. Visão Computacional. 3. Aprendizagem de Máquina. 4. Rastreamento Ocular. 5. Tecnologias Assistivas. I. Título.

CDD 006.3

ANNE LIVIA DA FONSECA MACEDO

**TÉCNICAS DE VISÃO COMPUTACIONAL APLICADAS NA DETECÇÃO E
RASTREAMENTO OCULAR PARA A INCLUSÃO DIGITAL DE PESSOAS COM
DEFICIÊNCIA MOTORA.**

Trabalho de Conclusão de Curso apresentado à
Faculdade de Computação da Universidade
Federal do Pará – Campus Castanhal, como
requisito parcial para obtenção do Título de
Bacharel em Sistemas de Informação.

Trabalho de Conclusão de Curso defendido e aprovado em: 13 de agosto de 2021.

Banca examinadora:

Prof. Dr. Igor Ruiz Gomes
(Orientador)
UFPA - Castanhal

Prof. Dr. José Jailton Henrique Ferreira Junior
UFPA - Castanhal

Prof. Dr. Tássio Costa de Carvalho
UFPA - Castanhal

Prof. Dr. Thiago Antônio Sidônio Coqueiro
UFPA - Castanhal

Castanhal - Pará
2021

Dedico este trabalho à minha mãe.

AGRADECIMENTOS

À minha família.

Ao Prof. Dr. Igor Ruiz Gomes.

À Faculdade de Computação.

À Universidade Federal do Pará.

“A arte é longa, a vida é breve.”

Hipócrates.

RESUMO

A despeito dos avanços significativos na tecnologia da informação, problemas relacionados à exclusão social persistem, com grande parte da população mundial sem acesso aos conteúdos digitais devido uma série de fatores econômicos, sociais e educacionais. Algumas das pessoas excluídas do universo digital são aquelas que apresentam dificuldades em movimentar os membros superiores, que normalmente são requisitados para o acionamento de determinados dispositivos computacionais como o teclado. Diante disso, o esforço para a inclusão digital tornou-se um consenso social, pois abrange a integração das pessoas à informática e a exploração de métodos que garantam a elaboração de tecnologias acessíveis a todos. Diferentes interfaces adaptadas estão sendo desenvolvidas para substituir os periféricos convencionais. Muitas dessas propostas baseiam-se no rastreamento ocular através de técnicas de visão computacional, propiciando a interação à distância, sem a necessidade de contato físico com o dispositivo. Fundamentando-se nesse contexto, o propósito principal desta pesquisa consistiu na implementação de uma tecnologia de baixo custo computacional voltada para pessoas com deficiência motora, que possibilita o acionamento das teclas direcionais ao rastrear o movimento dos olhos para quatro direções de interesse (cima, baixo, esquerda e direita). O sistema foi desenvolvido através da linguagem de programação C++ com o suporte dos algoritmos pré-implementados da biblioteca OpenCV, e concentrou-se no treinamento e aplicação do método de classificação Haar Cascade para a localização da região ocular e no uso de procedimentos específicos de visão computacional e processamento digital de imagens para determinar a direção do olhar com base no reconhecimento da pigmentação branca da esclerótica do globo ocular. Os resultados alcançados na fase experimental demonstram que o algoritmo proposto tem potencial suficiente para permitir o uso do sistema em tempo real, de forma apropriada e funcional. O modelo manifestou desempenho geral satisfatório na detecção dos olhos ao atingir uma taxa de 82% de verdadeiros positivos, 87% de precisão, 82% de sensibilidade e 84% de F1 Score, e foi capaz de reconhecer os movimentos oculares para as quatro direções de interesse, com tempo de resposta aceitável para execução de determinadas aplicações computacionais.

Palavras-Chave: Inclusão Digital. Visão Computacional. Aprendizagem de Máquina. Rastreamento Ocular. Tecnologias Assistivas.

ABSTRACT

Despite significant advances in information technology, issues related to social exclusion persist, where a large part of the world population does not have access to digital content due to a series of economic, social and educational factors. Some of the people excluded from the digital universe are those with upper limbs impairments, which are normally required to manipulate certain computational devices such as the keyboard. Therefore, digital inclusion efforts have gradually become a social consensus as it involves integrating people with technology and exploring effective methods to ensure the development of applications accessible to all. Different adapted interfaces are being developed to replace conventional peripherals. Many of these proposals are based on eye tracking using computer vision techniques, which conveniently provides distance interaction, without the need for physical contact with the device. Based on this context, the primary purpose of this research was to implement a low-cost computer technology aimed for people with limited mobility, which allows the control of the arrow keys by tracking the movement of the eyes in four directions of interest (up, down, left and right). The system was developed using the C++ programming language with the support of pre-implemented algorithms from the OpenCV library, and focused on the training and application of the Haar Cascade classification method for the localization of the eye region and in the usage of specific computer vision and digital image processing techniques to determine the direction of the gaze based on the recognition of the whiteness of the sclera. The results obtained in the experimental phase reveal that the proposed algorithm demonstrates sufficient potential to enable the real-time use of the system in an appropriate and functional manner. The model indicated satisfactory overall performance in the eye detection by achieving a true positive rate of 82%, accuracy of 87%, sensitivity of 82% and F1 score of 84% and managed to recognize the eye movements in the four directions of interest, with an acceptable response time for running some computing applications.

Keywords: Digital Inclusion. Computer Vision. Machine Learning. Eye Tracking. Assistive Technologies.

LISTA DE FIGURAS

Figura 01 – Estrutura do olho humano	21
Figura 02 – (a) Imagem contínua projetada em uma matriz de sensores, (b) resultado da amostragem e quantização	23
Figura 03 – Espectro eletromagnético	24
Figura 04 – Esquema do cubo de cores RGB	25
Figura 05 – Visualização geométrica do espaço de cores HSV	26
Figura 06 – Equalização do histograma de uma imagem com pouco contraste	27
Figura 07 – Operações morfológicas aplicadas sobre uma imagem	29
Figura 08 – Conjunto básico de características de Haar	31
Figura 09 – Extensão das características de Haar rotacionadas em 45°	31
Figura 10 – Exemplo de característica de Haar aplicada sobre uma imagem real	31
Figura 11 – Características que permitem o reconhecimento dos olhos e do nariz, respectivamente	32
Figura 12 – Imagem original à esquerda e imagem integral à direita	33
Figura 13 – Ilustração das áreas retangulares de uma imagem integral	33
Figura 14 – Ilustração do funcionamento do algoritmo AdaBoost	34
Figura 15 – Cascata de classificadores com N estágios	37
Figura 16 – Imagens positivas contendo olhos humanos	43
Figura 17 – Alguns exemplos do conjunto de imagens negativas	44
Figura 18 – Antes e o depois da remoção das áreas irrelevantes	47
Figura 19 – Padrão dos olhos ao olhar para a direita (a) e esquerda (b)	47
Figura 20 – Identificação do branco em diferentes distâncias, com moderada (a) e baixa (b) luminosidade	49
Figura 21 – Padrão dos olhos ao olhar direto (a) e para cima (b)	50
Figura 22 – Antes e o depois da aplicação da operação de dilatação	50
Figura 23 – Área da imagem analisada para apontar se o olhar está direcionado para cima	51
Figura 24 – Matriz de confusão para problemas de classificação binária	53
Figura 25 – Porcentagem de acertos e erros do classificador proposto e pré-treinado	55
Figura 26 – Detecção dos olhos sem falsos positivos e negativos	55
Figura 27 – Detecção dos olhos com a presença de alguns falsos positivos e negativos	56
Figura 28 – Aplicação desenvolvida para efetuar os experimentos no rastreamento dos olhos	56

Figura 29 – Identificação da cor branca para as quatro direções	57
Figura 30 – Sequência de movimentos para avaliar o tempo de execução	57
Figura 31 – Tempo de resposta para movimentar o personagem para à esquerda	58
Figura 32 – Tempo de resposta para movimentar o personagem para à direita	59
Figura 33 – Tempo de resposta para movimentar o personagem para cima	59
Figura 34 – Tempo de resposta para movimentar o personagem para baixo	60
Figura 35 – Tempo de resposta para movimentar o personagem nas quatro direções em conjunto	61
Figura 36 – Quantidade de acertos obtidos pelos dois usuários ao movimentar o personagem	62
Figura 37 – Identificação da coloração branca presente na esclera do primeiro voluntário .	62
Figura 38 – Pinguécula à esquerda e pterídeo à direita	63
Figura 39 – Identificação da coloração branca presente na esclera do segundo voluntário .	64

LISTA DE TABELAS

Tabela 1 – Configurações do computador	42
Tabela 2 – Limite inferior e superior da cor branca	46
Tabela 3 – Quantidade mínima de pixels para determinar o movimento na horizontal	48
Tabela 4 – Comparação entre os classificadores proposto e pré-treinado do OpenCV	54
Tabela 5 – Número de acionamentos para a conclusão de cada percurso	58

LISTA DE ALGORITMOS

Algoritmo 1 – AdaBoost	35
Algoritmo 2 – Algoritmo de treinamento do classificador em cascata	38

LISTA DE ABREVIATURAS E SIGLAS

AdaBoost	-	Adaptative Boosting
CIE	-	Commission Internationale de l'Éclairage
DPI	-	Dots Per Inch
ECG	-	Eletrocardiograma
EEG	-	Eletroencefalograma
EOG	-	Electrooculography
FDA	-	Função de Distribuição Acumulada
FN	-	Falso Negativo
FP	-	Falso Positivo
FPS	-	Frames Por Segundo
GUI	-	Interface Gráfica do Usuário
HSV	-	Hue, Saturation, Value
IBGE	-	Instituto Brasileiro de Geografia e Estatística
IHC	-	Interação Humano-Computador
IROG	-	Infrared Oculography
LBI	-	Lei Brasileira de Inclusão
LPI	-	Lines Per Inch
OMS	-	Organização Mundial da Saúde
OpenCV	-	Open Source Computer Vision Library
PNS	-	Pesquisa Nacional de Saúde
Pixel	-	Picture Element
RGB	-	Red, Green, Blue
ROI	-	Region of Interest
TA	-	Tecnologias Assistivas
VN	-	Verdadeiro Negativo
VOG	-	Video Oculography
VP	-	Verdadeiro Positivo
VR	-	Virtual Reality
XML	-	Extensible Markup Language

SUMÁRIO

1	INTRODUÇÃO	16
1.1	Objetivo geral	19
1.2	Objetivos específicos	19
1.3	Estrutura do trabalho	20
2	FUNDAMENTAÇÃO TEÓRICA	21
2.1	Olho humano	21
2.2	Processamento digital de imagens	22
2.2.1	Imagens digitais	22
2.2.2	Espaços de cores	24
2.2.3	Equalização do histograma	26
2.2.4	Operações morfológicas	28
2.3	Aprendizagem de máquina	29
2.4	Algoritmo de classificação Haar Cascade	30
2.4.1	Características de Haar	30
2.4.2	Imagens integrais	32
2.4.3	AdaBoost	34
2.4.4	Cascata de classificadores	36
3	TRABALHOS CORRELATOS	39
4	METODOLOGIA	42
4.1	Amostras positivas e negativas	43
4.2	Treinamento do classificador	44
4.3	Deteção dos olhos	45
4.4	Captura dos movimentos dos olhos	46
5	EXPERIMENTOS E RESULTADOS	52
5.1	Métricas de avaliação	52
5.2	Resultado da deteção dos olhos	53
5.3	Resultado do rastreamento dos olhos	56
6	CONSIDERAÇÕES FINAIS	65
	REFERÊNCIAS	67

1 INTRODUÇÃO

A informática teve sua origem no campo militar e rapidamente se alastrou pelo sistema produtivo, redesenhando os fluxos de capital desde a segunda metade do século XX, e recentemente modificou as formas de relacionamento social, os padrões culturais e a construção das subjetividades (CAZELOTO, 2019). As informações geradas atualmente estão sendo progressivamente armazenadas no formato digital, e sua manipulação advém exclusivamente por intermédio de máquinas com processadores específicos, como é o caso do computador (CARVALHO, 2012).

O acesso a essas informações é imprescindível para o desenvolvimento da sociedade. Entretanto, o avanço da informática ainda traz consigo diversos problemas relacionados à exclusão social, no qual uma grande parcela da população mundial acaba não tendo acesso aos conteúdos digitais devido fatores econômicos, sociais, educacionais, dentre outros (CARVALHO, 2012). Alguns dos excluídos do universo da cibercultura são os indivíduos com deficiências físicas ou cognitivas, principalmente aqueles com dificuldades em locomover os membros superiores (braço, antebraço, ombro e mão), que geralmente são requisitados para a utilização de determinadas ferramentas.

A usabilidade dos sistemas é um dos aspectos que impõe barreiras digitais. As técnicas genéricas para o acionamento desses dispositivos computacionais criados para pessoas sem deficiência abrangem mecanismos diretamente projetados para o uso das mãos ou dedos, como o teclado e o mouse. Para que uma pessoa impossibilitada de movimentar os membros superiores possa fazer o acionamento do computador, são necessários uma série de dispositivos especiais, como: próteses, órteses ou adaptações mecânicas (FRANÇA; BORGES; SAMPAIO, 2005).

Em conformidade com a Lei nº 13.146/15, são consideradas pessoas com deficiência, todas aquelas que possuem algum impedimento de longo prazo de natureza física, mental, intelectual ou sensorial, capaz de obstruir sua participação plena e efetiva na sociedade em igualdade de condições com as demais pessoas (BRASIL, 2015). Para o contexto deste trabalho, almeja-se facilitar o uso do computador por indivíduos com deficiência motora, contudo o sistema não se restringe somente a este grupo específico. Conforme Carvalho (2015), esse tipo de deficiência caracteriza-se por uma limitação parcial ou total do funcionamento do sistema musculoesquelético. Esta limitação é normalmente causada por fatores genéticos, infecções virais e bacterianas, ou traumas, que podem culminar em dificuldades na realização de atividades diárias.

Por iniciativa do Ministério da Saúde em convênio com o Instituto Brasileiro de Geografia e Estatística (IBGE), no ano de 2013 foi realizada no Brasil a Pesquisa Nacional de Saúde (PNS), com o objetivo principal de avaliar a situação de saúde e os estilos de vida dos brasileiros. Com base nos dados apresentados, 1,3% da população declarou possuir algum tipo de deficiência física, dentre esses indivíduos, 46,8% possuía grau intenso ou muito intenso de limitações (IBGE, 2015).

Segundo a Organização Mundial da Saúde (OMS), mais de um bilhão de pessoas convivem com algum tipo de deficiência, dentre os quais cerca de 200 milhões experimentam dificuldades funcionais consideráveis. As pessoas com deficiência apresentam normalmente piores expectativas de vida, níveis mais baixos de escolaridade, participação econômica menor, e taxas de pobreza mais elevadas se comparado às pessoas sem deficiência. Tais problemas originam-se dos obstáculos enfrentados em relação ao acesso a serviços como: saúde, educação e informação. Para alcançar melhores perspectivas, faz-se necessário capacitar as pessoas que vivem com deficiência e retirar as barreiras que as impedem de ter participação ativa na comunidade (OMS, 2011).

Diante dos fatos supracitados, os esforços para a realização da inclusão digital se tornaram um consenso social (CAZELOTO, 2019). Esta medida de inclusão ganha notoriedade por se tratar da integração das pessoas à tecnologia e da busca por métodos que propiciam o desenvolvimento tecnológico para que as inovações estejam acessíveis a todos (BARROS; SOUZA, 2020). Alicerçado neste contexto, surgem as tecnologias assistivas (TA), que podem proporcionar ou ampliar as habilidades funcionais das pessoas com deficiência, viabilizando independência, autonomia e inclusão (SANTOS *et al.*, 2017). O acesso à estas tecnologias é um direito previsto pela Lei Brasileira de Inclusão (LBI), nº 13.146/15 (BRASIL, 2015).

Consoante Rodrigues *et al.* (2018), a proposta da TA ainda é um grande desafio, em especial quando se trata da interação com o computador, uma vez que os dispositivos de interação precisam estar adaptados as necessidades dos usuários, caso contrário, esses artefatos podem não ser utilizados em toda sua potencialidade ou até mesmo nunca serem empregues. Atualmente, é a área de Interação Humano-Computador (IHC), que se preocupa com todos os aspectos relacionados ao processo de comunicação entre o ser humano e o sistema interativo, com o objetivo de desenvolver ou melhorar a segurança, utilidade, eficácia e usabilidade desses produtos (MAURI *et al.*, 2006).

Inúmeras estratégias podem ser desempenhadas para abrandar as dificuldades no uso desses equipamentos por parte das pessoas com deficiência. Segundo Azorín *et al.* (2014),

estão sendo desenvolvidos diferentes tipos de interfaces adaptadas ou equipamentos alternativos de interação, para substituir os periféricos convencionais como o teclado. Dentre essas interfaces encontram-se por exemplo, os mouses com apontador laser, mouses oculares, ou interfaces cérebro-computador. A proposta desta pesquisa envolve o acionamento das teclas direcionais através do rastreamento dos olhos.

Conforme Azorín *et al.* (2014), as interfaces baseadas em movimentos oculares utilizam a direção e o ângulo do olhar, de forma que pessoas com deficiência severa ou idosas possam utilizar tal interface para controlar desde um computador até dispositivos móveis como cadeiras de rodas e braços robóticos. Essas interfaces fundamentam-se em três princípios distintos:

- Video Oculography (VOG): baseia-se no registro da imagem do olho, mediante uma câmera.
- Infrared Oculography (IROG): funda-se em registrar o reflexo de luz infravermelha sobre o olho através de uma câmera ou um detector de infravermelho.
- Electrooculography (EOG): consiste em registrar, por meio de eletrodos superficiais, a mudança de potencial produzido no olho durante a realização dos movimentos oculares.

O processo de desenvolvimento desta pesquisa seguiu os princípios da técnica VOG, que em concordância com Azorín *et al.* (2014), norteia-se na utilização de uma câmera de vídeo e técnicas de processamento de imagens e visão computacional para avaliar a direção do olhar. Para Mendonça (2019), a visão computacional é uma área da ciência da computação que, aliada a inteligência artificial, pode dar ao computador a capacidade de interpretar uma imagem do mundo real, extraíndo e analisando diversas informações que permitam auxiliar o usuário. Com o avanço da tecnologia, os algoritmos de visão computacional têm demonstrado ser cada vez mais eficientes para detectar aspectos visuais e comportamentais (SOUZA, 2019).

Segundo Mauri *et al.* (2006), os sistemas de interação baseados em visão computacional são extremamente flexíveis, pois qualquer modificação detectada no vídeo pode ser interpretada pelo computador e usada para desencadear alguma ação. Dentre as vantagens existentes, se encontram: a possibilidade do usuário interagir à distância, sem contato físico com o dispositivo; a oportunidade de combinar periféricos clássicos de interação, como o teclado ou o mouse, para melhorar a interação de forma multimodal;

además, tecnologías baseadas em webcam, por exemplo, são particularmente mais acessíveis, se comparado aos dispositivos de visão computacional profissionais (industriais).

Dessarte, fundamentando-se na problemática exposta, esta pesquisa contribui com projetos na área, através do desenvolvimento de uma tecnologia direcionada a pessoas com deficiência motora, por meio da aplicação de técnicas de baixo custo computacional pertencentes aos campos de aprendizagem de máquina, processamento digital de imagens e visão computacional. O método proposto é executado em duas etapas: primeiramente é efetuada a detecção dos olhos, e em seguida são impostos procedimentos específicos para a extração e análise dos movimentos oculares, com o intuito de obter comandos direcionais válidos e aplicá-los ao controle das teclas direcionais.

1.1 Objetivo geral

Utilizar algoritmos de visão computacional, aprendizagem de máquina e processamento digital de imagens, no desenvolvimento de um processo de software de baixo custo computacional, prático (não requer calibração e inserção de dados manualmente), viável (tempo de resposta apropriado) e não invasivo (sem exigir contato direto com o globo ocular), capaz de efetuar a detecção e o rastreamento dos olhos do usuário, para posteriormente aplicar as ações processadas, no controle das teclas direcionais. O sistema proposto deve auxiliar pessoas com deficiência motora, na execução de diversas tarefas computacionais, que necessitam do movimento de um cursor para uma direção específica, como é o caso de alguns jogos eletrônicos e tecnologias de realidade virtual (*Virtual Reality - VR*).

1.2 Objetivos específicos

Para alcançar o objetivo geral do trabalho proposto, foram destacados como objetivos específicos:

- Estudo de técnicas de inteligência artificial para descoberta e análise de padrões em imagens digitais e verificação das que mais se adequam para o desenvolvimento de uma metodologia de rastreamento dos olhos, considerando as diferentes características étnico-raciais e as diversas circunstâncias ambientais em que o sistema pode ser utilizado;
- Efetuar a detecção dos olhos em tempo real através da webcam, utilizando uma abordagem de aprendizagem de máquina de baixo custo e eficiente;

- Identificar a direção do olhar através de técnicas de processamento digital de imagens e visão computacional, levando em consideração as quatro direções de interesse: cima, baixo, esquerda e direita;
- Aplicar os comandos obtidos no controle das teclas direcionais;
- Desenvolver uma aplicação com interface gráfica amigável, que possua funcionalidades que permitam validar os resultados através de experimentos que mensuram o tempo demandado para acionar uma direção específica.

1.3 Estrutura do trabalho

Este trabalho encontra-se dividido em 6 capítulos. Neste primeiro capítulo introdutório, são destacadas a problemática e a proposta da pesquisa, os objetivos gerais e específicos. O restante dos capítulos está organizado como segue.

No segundo capítulo são apresentadas as fundamentações teóricas que embasaram esta pesquisa, destacando os conceitos referentes à estrutura do olho humano, bem como o funcionamento das técnicas computacionais utilizadas durante o desenvolvimento, como: operações morfológicas, espaços de cores e algoritmos de aprendizagem de máquina, com ênfase principal no método de detecção de objetos *Haar Cascade*.

No terceiro capítulo são apresentados alguns trabalhos correlatos ao tema abordado, envolvendo diferentes metodologias de desenvolvimento.

No quarto capítulo é sumarizada a metodologia empregada na pesquisa e descrita as etapas seguidas para o desenvolvimento do sistema, contendo informações técnicas acerca da aquisição das imagens para treinamento do classificador e dos procedimentos implementados em C++ para alcançar os objetivos propostos.

O quinto capítulo trata dos resultados obtidos durante a etapa de realização dos experimentos, destacando uma comparação de desempenho entre o classificador treinado proposto por esta pesquisa e o classificador pré-treinado pertencente à biblioteca *OpenCV*. Além disso, são apresentados os resultados referentes à detecção da direção do olhar, por meio da verificação do tempo demandado para acionar as quatro direções de interesse: cima, baixo, esquerda e direita.

Por fim, no sexto capítulo são apresentadas as considerações finais e elencadas algumas propostas para trabalhos futuros, que permitam aprimorar ainda mais o desempenho do sistema proposto em diferentes aspectos, principalmente relacionados com a usabilidade, precisão e praticidade.

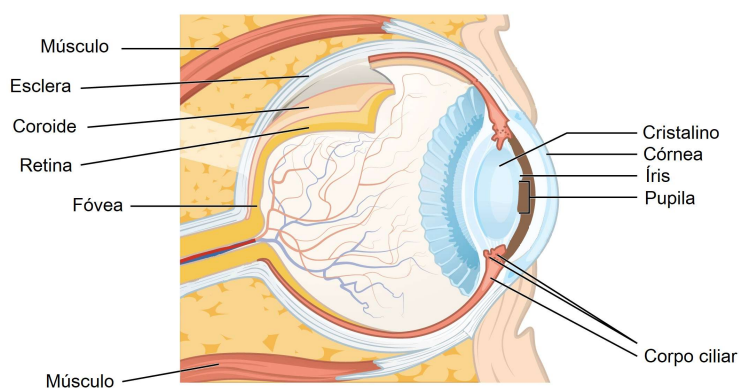
2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são abordados os pressupostos teóricos referentes à estrutura do olho humano e as técnicas de processamento digital de imagens, visão computacional e aprendizagem de máquina, que contribuíram para o desenvolvimento deste trabalho.

2.1 Olho humano

Os olhos humanos são órgãos que possuem formato aproximadamente esférico e um diâmetro de cerca de 20mm, que em essência, é um receptor de luz que consegue converter energia luminosa em impulsos elétricos para serem decodificados pelo sistema nervoso (RAMOS, 2006). Este órgão é revestido por três membranas: a córnea e a esclerótica que formam a camada externa, a coroide e a retina. A Figura 1 é um exemplo da estrutura do olho humano.

Figura 1 – Estrutura do olho humano.



Fonte: Adaptado de Biga (2019).

A córnea é um tecido resistente e transparente que cobre a superfície anterior do olho, e como um prolongamento da córnea, a esclera é uma membrana opaca que reveste o restante do globo ocular (GONZALEZ; WOODS, 2009) e possui os músculos responsáveis pelos movimentos dos olhos (CAPELINI, 2019). Esses músculos são conhecidos como: reto superior, reto medial, reto inferior, reto lateral, oblíquo superior e oblíquo inferior (MAIA, 2019).

A coroide situa-se abaixo da esclera, se divide em corpo ciliar e íris, e possui revestimento substancialmente pigmentado, que ajuda, através da pupila, a reduzir a quantidade de luz indesejável que entra no olho e se espalha pelo globo ocular (GONZALEZ; WOODS, 2009). Além disso, a coroide é responsável por fornecer oxigênio e nutrientes para as células da retina (CAPELINI, 2019).

A entrada da luz se dá pela córnea, atinge a pupila e em seguida chega ao cristalino, que atua como uma lente de focalização responsável por fazer os raios convergirem em um ponto chamado retina (MANZI, 2007). A retina é a membrana capaz de transformar os impulsos nervosos em percepções visuais através dos receptores de luz discretos conhecidos como: cones e bastonetes. Os cones estão localizados na porção central da retina, chamada de fóvea e são encarregados de fornecer a percepção das cores, já os bastonetes são sensíveis a baixa intensidade luminosa e são responsáveis por dar uma visão geral da imagem captada no campo de visão (MARQUES FILHO; VIEIRA NETO, 1999).

2.2 Processamento digital de imagens

A área de processamento digital de imagens tem apresentado um crescimento expressivo nos últimos anos, permitindo a solução de uma ampla variedade de problemas nas mais diversas áreas. O grande interesse pelos métodos da área provém de duas categorias distintas: o aprimoramento das informações para a interpretação humana; e a extração das informações para a análise automática por máquinas (MARQUES FILHO; VIEIRA NETO, 1999). Os subtópicos seguintes detalham alguns dos conceitos referentes ao processamento digital de imagens, que foram úteis no decorrer da pesquisa.

2.2.1 Imagens digitais

Uma imagem pode ser definida como uma função bidimensional $f(x, y)$, em que x e y são consideradas coordenadas espaciais (plano), e a amplitude de f em qualquer par de coordenadas (x, y) , representa a intensidade ou o nível de cinza neste ponto específico da imagem. Para a imagem ser considerada uma imagem digital, tanto o nível de intensidade, quanto as coordenadas x e y devem possuir valores discretos e finitos (GONZALEZ; WOODS, 2009).

Segundo Marques Filho e Vieira Neto (1999), a intensidade de uma imagem pode ser caracterizada fisicamente através da combinação de dois componentes: a iluminância $i(x, y)$, que se refere à quantidade de luz que incide na cena que está sendo vista, e a refletância $r(x, y)$, que indica a quantidade de iluminação refletida pelos objetos presentes na imagem. Portanto, $f(x, y)$ é representado matematicamente como:

$$f(x, y) = i(x, y) * r(x, y) \quad (1)$$

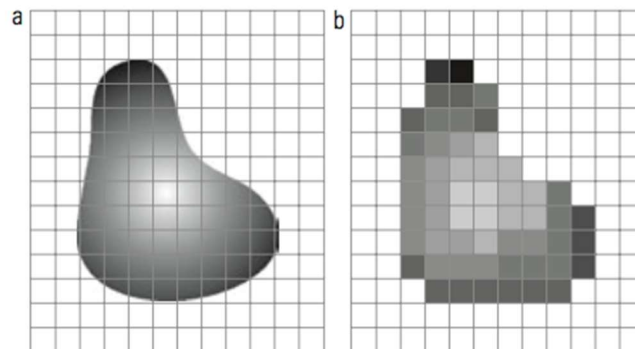
Sendo:

$$0 < i(x, y) < \infty \quad \text{e} \quad 0 < r(x, y) < 1$$

A função $i(x, y)$ é medida em lúmen/m² ou lux, e a função $r(x, y)$ apresenta valores percentuais entre 0 (absorção total) e 1 (refletância total). A natureza de $i(x, y)$ é determinada pela fonte de iluminação, enquanto que $r(x, y)$ é determinada pelas características dos objetos presentes na cena, como sua textura e coloração (GONZALEZ; WOODS, 2009).

Computacionalmente, para obter a imagem digital, é necessário submetê-la ao processo de digitalização, que é realizado em dois passos: a amostragem, que consiste em discretizar o domínio de definição nas direções x e y , gerando uma matriz de dimensões $M \times N$; e em seguida é efetuada a quantização, que diz respeito a escolha do valor inteiro L de níveis de cinza para cada ponto da imagem (CHIBA, 2014). Cada célula dessa matriz é chamada de pixel (*Picture Element*). O resultado do processo de digitalização de uma imagem é ilustrado pela Figura 2.

Figura 2 – (a) Imagem contínua projetada em uma matriz de sensores, (b) resultado da amostragem e quantização.



Fonte: Gonzalez e Woods (2009, p. 35).

Conforme Burger e Burge (2009), a informação presente em um elemento da imagem depende do tipo de dado utilizado para representá-lo, e seus valores são dependentes da profundidade K (quantidade de bits) de cada pixel, ou seja, cada elemento pode representar qualquer um dos 2^k valores diferentes. Existem três tipos de imagens mais comuns: a imagem binária, a imagem de intensidade e a imagem colorida. Todos os três tipos de imagens foram importantes no desenvolvimento deste trabalho.

A imagem binária é um tipo especial de imagem em escala de cinza, onde cada pixel pode assumir apenas um de dois valores distintos: branco ou preto. Uma imagem em tons de cinza consiste de uma matriz que apresenta números inteiros normalmente de 8 bits, cuja intensidade ou densidade é indicada através de valores que variam de $L_{min} = 0$ a $L_{max} = 255$, onde 0 representa a coloração preta e o valor 255 diz respeito a cor branca (GONZALEZ;

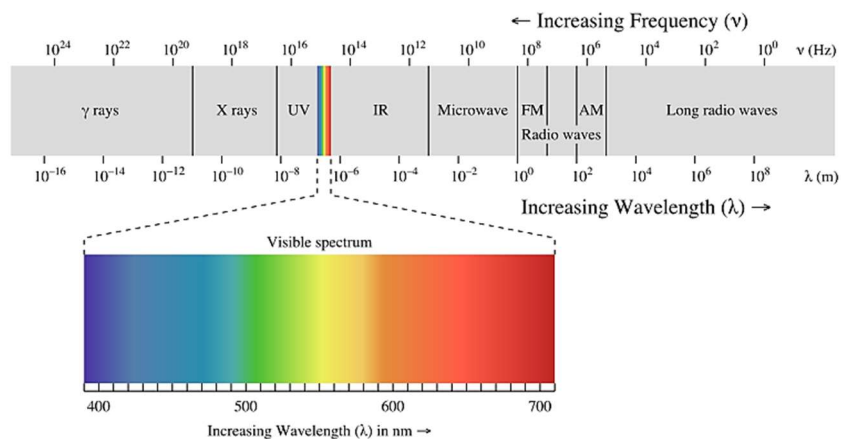
WOODS, 2009). Já a imagem colorida é formada através da combinação de uma sequência de imagens monocromáticas, como é o caso das imagens padrão RGB (*Red, Green and Blue*), que possuem três bandas, cada uma com sua própria função $f(x, y)$ (CHIBA, 2014).

A qualidade da imagem digital tem relação direta com sua resolução, que especifica as dimensões espaciais da imagem no mundo real, através do número de elementos presentes na imagem por medida como *dots per inch (dpi)*, *lines per inch (lpi)* ou pixels por quilômetros quando se trata de imagens de satélites (BURGER; BURGE, 2009). Em geral, quanto maior for a quantidade de pixels por polegada, melhor será a qualidade da imagem.

2.2.2 Espaços de cores

As colorações nas imagens digitais são poderosos descritores que muitas das vezes simplifica a identificação e segmentação de objetos (GONZALEZ; WOODS, 2009). A cor do ponto de vista físico se refere, na realidade, à uma manifestação perceptual da luz que, por sua vez, é considerada uma radiação eletromagnética, que possui uma certa energia para cada comprimento de onda. A função que informa o valor dessa energia é chamada de distribuição espectral de potência (SCURI, 1999). A Figura 3 ilustra o espectro de radiações eletromagnéticas, destacando a faixa visível ao olho humano, que compreende radiações com comprimentos de ondas entre 400nm a 700nm, variando do violeta em uma extremidade, ao vermelho na outra.

Figura 3 – Espectro eletromagnético.



Fonte: Volchko (2018).

As cores que percebemos são geradas a partir de diversos processos psicoquímicos, que podem ser divididos em três grandes grupos: o processo aditivo, onde vários raios de luz são combinados para formar um novo raio; o processo subtrativo, em que um raio de luz passa por um filtro que elimina alguns comprimentos de ondas resultando em uma nova cor; e por

fim, o processo por pigmentação, que gera uma nova cor após a luz passar por diversos eventos, sendo refletida, transmitida e absorvida. O processo aditivo é utilizado em monitores de computadores e televisões, o processo subtrativo aparece em projetores, e o processo por pigmentação está presente em pinturas e trabalhos impressos (SCURI, 1999).

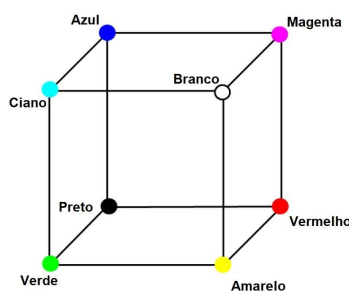
Para realizar a reprodução das cores é suficiente, na maioria das vezes, considerar a luz branca como uma mistura de apenas três componentes luminosos: as luzes vermelha, verde e azul, conhecidas como cores primárias aditivas (CHIBA, 2014), devido a possibilidade de obter qualquer outra cor através da combinação aditiva de uma ou mais delas, em diferentes proporções.

A mistura das cores primárias duas a duas, produzem as chamadas cores secundárias ou cores primárias subtrativas, que são o magenta (vermelho + azul), amarelo (vermelho + verde), e o ciano (verde + azul) (MARQUES FILHO; VIEIRA NETO, 1999). A luz branca é obtida através da combinação das três cores primárias, enquanto que a união das três cores secundárias produz a cor preta, indicando a ausência de luz (CHIBA, 2014).

Os espaços de cores foram criados para facilitar e padronizar a caracterização das colorações. Esses modelos são basicamente representações tridimensionais, em que cada cor é indicada por um ponto específico no sistema de coordenadas 3D (MARQUES FILHO; VIEIRA NETO, 1999). O órgão responsável por essa padronização é a Comissão Internacional de Iluminação (CIE, *Commission Internationale de l'Éclairage*), e alguns dos modelos mais utilizados são o RGB e o HSV.

O modelo RGB utiliza as cores primárias aditivas para representá-las em dispositivos como os monitores e as câmeras de vídeo (CHIBA, 2014). Este modelo é baseado em um sistema de coordenadas cartesianas, que conforme ilustrado pela Figura 4, pode ser visto como um cubo onde, três de seus vértices são as cores primárias, outros três são as cores secundárias, o vértice junto à origem é o preto e o mais afastado da origem corresponde à cor branca (MARQUES FILHO; VIEIRA NETO, 1999).

Figura 4 – Esquema do cubo de cores RGB.

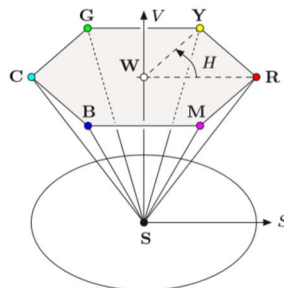


Fonte: Adaptado de Westland e Cheung (2015).

O modelo HSV (*hue, saturation and value*) é utilizado de forma bastante intensa em diversos sistemas de visão artificial fortemente baseados no modelo de percepção de cor pelo ser humano (MARQUES FILHO; VIEIRA NETO, 1999). Neste padrão, as cores são especificadas através dos componentes matiz (*hue*), saturação (*saturation*) e valor ou brilho (*value*). O matiz é o atributo que descreve a cor em si e define a tonalidade dominante de uma área (ALVES, 2010). A saturação fornece a medida da diluição da cor pura por luz branca (WATANABE, 2019), por exemplo, enquanto que o vermelho é a cor pura, a tonalidade rosa é um vermelho com alguma quantidade de cor branca. Já o componente valor refere-se à luminância da cor, que varia de 0 (preto) ao 1 (branco) ao longo do eixo V da pirâmide, onde se encontra a escala de cinza (ALVES, 2010).

Conforme apresentado pela Figura 5, o HSV pode ser tradicionalmente visualizado como uma pirâmide de seis lados invertida, em que o eixo vertical representa o valor de V (brilho), a distância horizontal do eixo é o valor de S (saturação) e o ângulo representa o valor de H (matiz). O ponto preto está na ponta da pirâmide e o ponto branco fica no centro da base. Já as três cores primárias vermelho, verde e azul e as cores secundárias amarelo, ciano e magenta são os pontos de canto da base (BURGER; BURGE, 2009).

Figura 5 – Visualização geométrica do espaço de cores HSV.



Fonte: Burger e Burge (2009, p. 207).

2.2.3 Equalização do histograma

Para Marques Filho e Vieira Neto (1999), o histograma de uma imagem descreve a frequência dos valores de intensidade que ocorrem na cena. Esses valores são normalmente representados através de um gráfico de barras que fornece para cada nível de cinza, o número ou percentual de pixels correspondentes na imagem. Através da visualização do histograma, é possível verificar a qualidade da imagem quanto ao nível de contraste e também de brilho.

Matematicamente, o histograma normalizado de uma imagem na escala de cinza, com valores de intensidade no intervalo de 0 a $L - 1$, é obtido através da fórmula:

$$p(r_k) = \frac{n_k}{n} \quad (2)$$

Onde, r_k é o k -ésimo valor de intensidade, n_k é o número de pixels da imagem com intensidade r_k e n é o número total de pixels da imagem. $p(r_k)$ é uma estimativa da probabilidade de ocorrência do nível de intensidade r_k em uma imagem. Portanto, a soma de todos os componentes de um histograma normalizado deve ser igual a 1, em conformidade com os axiomas e teoremas da teoria da probabilidade.

Usualmente empregada como uma das etapas de pré-processamento em aplicações que envolve o reconhecimento de padrões, a equalização do histograma é uma técnica que visa ajustar a escala de cinza da imagem para obter um histograma uniforme (MARQUES FILHO; VIEIRA NETO, 1999), o que permite realçar diferenças de tonalidade, provocando um aumento significativo no nível de detalhes perceptíveis (QUEIROZ; GOMES, 2006).

Segundo Marques Filho e Vieira Neto (1999), uma das formas mais usuais de se equalizar um histograma é através da função de distribuição acumulada (FDA) da distribuição de probabilidades original, que pode ser expressa por:

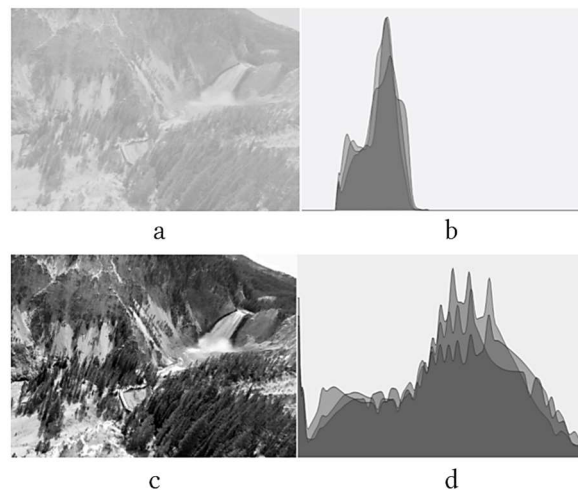
$$s_k = T(r_k) = \sum_{j=0}^k \frac{n_j}{n} = \sum_{j=0}^k p_r(r_j) \quad (3)$$

Sendo:

$$0 \leq r_k \leq 1 \quad \text{e} \quad k = 0, 1, \dots, L - 1$$

A Figura 6 é um exemplo da aplicação da equalização do histograma, onde a letra (a) retrata uma imagem com pouco contraste, conforme consta no histograma da letra (b), e a letra (c) exibe a mesma imagem após a aplicação da técnica, que acabou resultando no histograma da letra (d), com os níveis de intensidade melhores distribuídos. Conforme Tripathi, Mukhopadhyay e Dhara (2011), histogramas largos refletem uma cena com contraste significativo, enquanto que histogramas estreitos indicam menos contraste.

Figura 6 – Equalização do histograma de uma imagem com pouco contraste.



Fonte: Adaptado de Kaehler e Bradski (2016, p. 329 e 331).

2.2.4 Operações morfológicas

As operações morfológicas são comumente utilizadas no pré e pós processamento de imagens por possibilitar a remoção de ruídos através do espessamento e afinamento dos objetos, o que permite unificar os componentes de uma grande região, que poderiam estar separados em múltiplas partes, devido sombras ou outros efeitos similares (KAEHLER; BRADSKI, 2016). Além disso, essas operações podem ser aplicadas em diversos outros processamentos como: segmentação, detecção de bordas e esqueletização (MARQUES FILHO; VIEIRA NETO, 1999).

Conforme Queiroz e Gomes (2006), o modelo morfológico para a análise de imagens, fundamenta-se na extração das informações mediante transformações morfológicas, nos conceitos da álgebra booleana e na teoria dos conjuntos e reticulados. Este princípio se embasa no fato de que a imagem é um conjunto de pontos elementares (pixels), que originam subconjuntos elementares bi ou tridimensionais. Os subconjuntos e a inter-relação entre eles formam estruturalmente a morfologia da imagem. As duas operações mais comuns e fundamentais para o processamento morfológico são: Erosão e Dilatação.

A erosão quando aplicada provoca o afinamento dos objetos presentes na imagem binária (GONZALEZ; WOODS, 2009), a partir da remoção dos pixels que não satisfazem um determinado padrão (QUEIROZ; GOMES, 2006). A erosão de A (imagem binária) por B (elemento estruturante), sendo A e B conjuntos de Z^2 , indicada por $A \ominus B$, tem como uma de suas definições:

$$A \ominus B = \{z \mid (B)_z \subseteq A\} \quad (4)$$

O que indica que a erosão de A por B resulta no conjunto de pontos z, tal que B, transladado por z, está contido em A. Para Gonzalez e Woods (2009), os elementos estruturantes são considerados pequenos conjuntos utilizados para examinar uma imagem buscando propriedades de interesse, e a translação de um conjunto B no ponto $z = (z_1, z_2)$, indicada por $(B)_z$, é definida como:

$$(B)_z = \{c \mid c = b + z, b \in B\} \quad (5)$$

Segundo Gonzalez e Woods (2009), ao contrário da erosão, a dilatação provoca o espessamento dos objetos contidos na imagem binária. A forma específica e a extensão desse espessamento são controladas pelo formato do elemento estruturante. A equação da dilatação, denotada por $A \oplus B$, sendo A e B conjuntos no espaço Z^2 , tem como uma de suas definições:

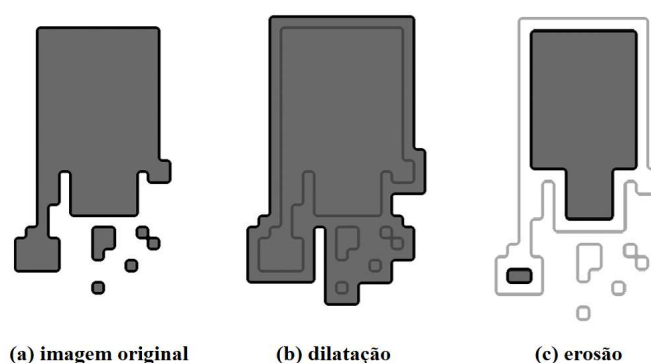
$$A \oplus B = \{z \mid [(B)_z \cap A] \subseteq A\} \quad (6)$$

Indicando que o processo de dilatação se baseia na reflexão de B em torno de sua origem, seguida da translação dessa reflexão por z. O conceito da reflexão de B, denotado como \hat{B} , é indicado como:

$$\hat{B} = \{w \mid w = -b, \text{ para } b \in B\} \quad (7)$$

Portanto, a dilatação de A por B é o conjunto de todos os deslocamentos z, de forma que \hat{B} e A se sobreponham por pelo menos um elemento. A Figura 7 ilustra o resultado da aplicação das operações de erosão e dilatação sobre uma imagem.

Figura 7 – Operações morfológicas aplicadas sobre uma imagem.



Fonte: Adaptado de Kaehler e Bradski (2016, p. 277)

2.3 Aprendizagem de máquina

A aprendizagem de máquina (*Machine Learning*) é um ramo da inteligência artificial que objetiva permitir que as máquinas realizem seus trabalhos, utilizando softwares inteligentes baseados em métodos estatísticos (MOHAMMED; KHAN; BASHIER, 2016). Estes sistemas tomam decisões alicerçadas pelas experiências acumuladas através das soluções bem-sucedidas dos problemas anteriores (MONARD; BARANAUSKAS, 2003). Essas soluções são baseadas no raciocínio indutivo, que extrai regras e padrões de um determinado conjunto de dados, para conceder uma descrição útil de algo. O raciocínio indutivo pode ser subdividido em dois dos principais métodos de aprendizagem: o supervisionado e o não supervisionado (CORCOVIA; ALVES, 2019).

O aprendizado supervisionado ocorre quando são fornecidos ao sistema um conjunto de exemplos com suas respectivas classes para, através deste conjunto, classificar os outros conjuntos de dados ainda não rotulados. Quando a previsão abrange valores numéricos contínuos, temos a regressão, ao passo que, se a previsão estiver relacionada a classes discretas e pré-definidas, temos a chamada tarefa de classificação (MACHADO, 2017). Alguns dos algoritmos de aprendizagem supervisionada existentes são: Árvores de decisão

(*Decision Trees*), Floresta aleatória (*Random Forest*), K-Vizinhos mais próximos (*K-Nearest Neighbors*) e Regressão logística (*Logistic Regression*).

O aprendizado de máquina não supervisionado supõe que a amostra de treinamento não possui rótulos, com isto o objetivo do sistema consiste de observar a similaridade entre os dados, para então agrupar os exemplos com características semelhantes em determinadas classes (MACHADO, 2017). Entre os algoritmos dessa categoria se encontram: o Agrupamento por K-Médias (*K-Means Clustering*) e o Algoritmo Apriori (*Apriori Algorithm*).

O algoritmo de aprendizagem selecionado para localizar a região dos olhos em tempo real, faz uso de procedimentos de classificação supervisionada. Os tópicos seguintes detalham os conceitos primordiais para o funcionamento deste detector.

2.4 Algoritmo de classificação Haar Cascade

O método *Haar Cascade* é intensamente explorado na literatura e tem como fundamento a abordagem de aprendizagem de máquina proposta por Viola e Jones (2001), em que uma função em cascata é treinada a partir de muitas imagens positivas e negativas, para posteriormente efetivar a detecção de objetos em outras imagens (RYUMIN *et al.*, 2020). Essa abordagem foi inicialmente projetada para a detecção de faces em tempo real, no entanto, diversos experimentos comprovam que este modelo computacional pode ser utilizado de maneira altamente eficiente para a detecção de outros objetos.

Segundo Pereira (2017), esse algoritmo além de proporcionar grande liberdade de escolha do alvo de acordo com a aplicação a ser desenvolvida, por dispor de técnicas de treinamento inteligente e por fornecer procedimentos rápidos e de baixo custo computacional para a detecção em tempo real, é considerado um classificador de grande robustez, por permitir o treinamento para a captura de alvos em circunstâncias adversas como: baixa iluminação e oclusão parcial do objeto. Quatro conceitos fundamentais são introduzidos por este método: as características de Haar, as imagens integrais, o algoritmo de aprendizado supervisionado *AdaBoost* e os classificadores em cascata.

2.4.1 Características de Haar

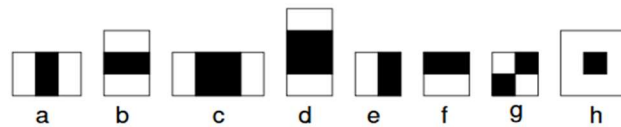
As características de Haar (*Haar-Like Features*) foram inspiradas no trabalho de Papageorgiou, Oren e Poggio (1998), e são comumente aplicadas na etapa de extração de padrões e texturas, para serem utilizadas como entrada do classificador em cascata. Esses

descritores são constituídos de dois ou mais retângulos adjacentes de coloração branca e preta. Cada uma dessas características transforma-se em um único valor obtido através da diferença entre a soma de todos os pixels nos retângulos brancos, pela soma de todos os pixels nos retângulos pretos (GONZALEZ; VELÁSQUEZ, 2019). Conforme Silva (2018), este processo pode ser matematicamente descrito como:

$$\Delta = \frac{1}{n} \sum_{preto}^n I(x) - \frac{1}{n} \sum_{branco}^n I(x) \quad (8)$$

A Figura 8 ilustra algumas das características de Haar básicas, onde as letras (a), (b), (c) e (d) são características de linha, as letras (e) e (f) são características de borda, a letra (g) é a característica de linha diagonal e a letra (h) é a característica *center-surround*.

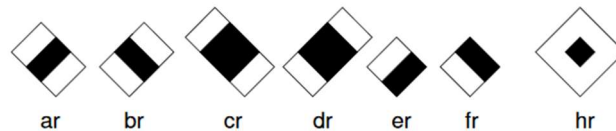
Figura 8 – Conjunto básico de características de Haar.



Fonte: Belussi e Hirata (2011).

Com o propósito de aprimorar a performance na detecção, foi proposta por Lienhart e Maydt (2002), uma extensão desse conjunto, onde foi incluída as características rotacionadas em 45°, conforme apresentado pela Figura 9. Através desta extensão, foi possível reduzir em média, até 10% o índice de falsos alarmes.

Figura 9 – Extensão das características de Haar rotacionadas em 45°.



Fonte: Belussi e Hirata (2011).

A Figura 10 é um exemplo da aplicação de uma característica de Haar com dimensões de 4x4 pixels, sobre uma imagem real. Através da equação 8, obtém-se: $\Delta = 0.74 - 0.18 = 0.56$. Quanto mais próximo de 1, mais semelhante é a imagem real da característica aplicada (SILVA, 2018).

Figura 10 – Exemplo de característica de Haar aplicada sobre uma imagem real.

0	0	1	1	0.2	0.2	0.8	0.6
0	0	1	1	0.1	0.3	0.6	0.8
0	0	1	1	0.2	0.1	0.8	0.8
0	0	1	1	0.2	0.1	0.6	0.9

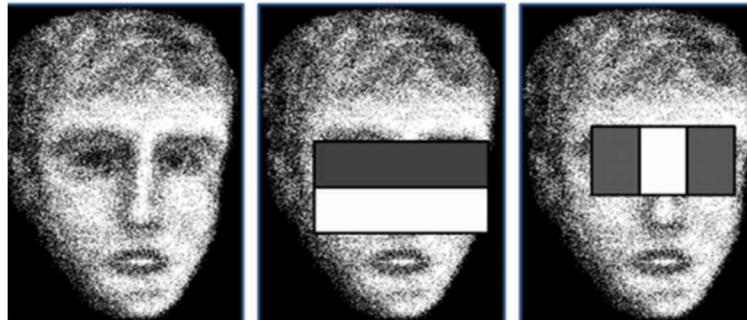
(a) Intensidade ideais (b) Valores reais

Fonte: Adaptado de Silva (2018).

Segundo Silva (2018), umas das motivações para a utilização das características de Haar se dá pela razão de que trabalhar com essas características é mais vantajoso do que trabalhar com cada pixel da imagem separadamente, devido a possibilidade de reconhecer determinados padrões e ao baixo custo computacional, pois conforme Moreira (2008), o número de características é substancialmente inferior em relação a quantidade de pixels de um objeto, o que permite que a análise da imagem seja efetuada com maior velocidade.

Conforme exemplificado pela Figura 11, a característica da Figura 8f, pode ser empregada no reconhecimento da região dos olhos, que costuma ser mais escura do que a região das bochechas, assim como a característica da Figura 8a pode exibir a distinção entre as intensidades da região dos olhos e do nariz.

Figura 11 – Características que permitem o reconhecimento dos olhos e do nariz, respectivamente.



Fonte: Alling, Powers e Soyata (2016).

2.4.2 Imagens integrais

O número de características que podem ser manipuladas em um classificador robusto e real é bastante elevado, portanto, o processamento direto da soma dos pixels sobre múltiplos retângulos, passaria a ser uma tarefa bastante custosa e não apropriada para aplicações em tempo real (MONTEIRO; PEIXOTO; NUNES, 2006). Com a intenção de aumentar a eficiência neste cálculo, foi proposta por Viola e Jones (2001), a utilização de uma representação intermediária da imagem original, conhecida por imagem integral, que permite que os descritores possam ser facilmente computados em qualquer escala e localização em tempo constante, usando apenas algumas operações por pixel.

Na imagem integral, cada pixel na posição (x, y) , armazena a soma de todos os pixels situados acima e à esquerda do pixel em questão na imagem original. Este cálculo é representado matematicamente como:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (9)$$

Onde $ii(x, y)$ é a imagem integral e $i(x, y)$ representa a imagem original. Este processamento pode ainda ser efetuado em apenas uma única varredura sobre a imagem, através dos pares de recorrência:

$$s(x, y) = s(x, y - 1) + i(x, y) \quad (10)$$

$$ii(x, y) = ii(x - 1, y) + s(x, y) \quad (11)$$

Sendo $s(x, y)$ a soma acumulada de uma linha e $s(x, -1)$ e $ii(-1, y)$ definidos com o valor 0. O resultado do processo de transformação de uma imagem de entrada para a imagem integral pode ser demonstrado através da Figura 12.

Figura 12 – Imagem original à esquerda e imagem integral à direita.

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

Imagem de entrada

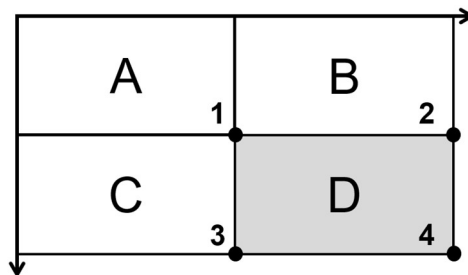
1	2	3	4
2	4	6	8
3	6	9	12
4	8	12	16

Imagem integral

Fonte: Elaborada pela autora.

Após a obtenção da imagem integral, as operações podem ser efetuadas a partir da soma dos vértices de uma região retangular em $ii(x, y)$, representando a soma de todos os pixels situados na área retangular equivalente da imagem original (ZAGHETTO *et al.*, 2017). Conforme Viola e Jones (2001), a soma de qualquer área retangular é alcançada através de somente 4 valores. À vista disso, o tempo de processamento requerido para computar as características não é dependente do seu tamanho (UDDIN; AKHI, 2016). A Figura 13 exemplifica este cálculo através de quatro áreas retangulares nomeadas de A, B, C e D, definidas pelos pontos 1, 2, 3 e 4.

Figura 13 – Ilustração das áreas retangulares de uma imagem integral.



Fonte: Adaptado de Macedo e Gomes (2020).

Nesta imagem, o ponto 1 equivale à soma de todos os pixels no retângulo A, o ponto 2 diz respeito à soma dos pixels nos retângulos A e B, o ponto 3 representa a soma dos pixels

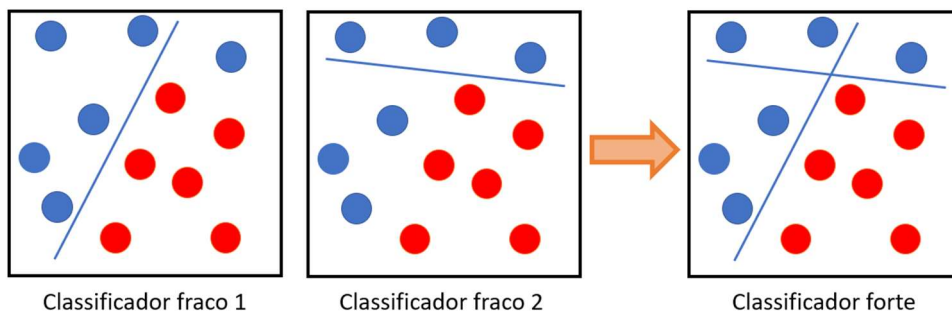
situados em A e C, e o ponto 4 armazena o somatório de todos os pixels existentes nas 4 regiões retangulares. Dessa forma, o cálculo necessário para obter a soma dos pixels no retângulo D seria: $4 - (2 + 3) + 1$. O ponto 1 deve ser acrescentado, pois seu valor é subtraído duas vezes, visto que ambos os pontos 2 e 3 incluem o retângulo A em sua soma.

2.4.3 AdaBoost

Com o objetivo de propiciar uma rápida classificação, dado um conjunto de características e um conjunto de imagens positivas e negativas para treinamento, foi proposta por Viola e Jones (2001), a utilização de uma variante do algoritmo de aprendizagem de máquina supervisionado *AdaBoost* (*Adaptive Boosting*). No sistema desenvolvido pelos pesquisadores, o *AdaBoost* é aplicado tanto para treinar o classificador, quanto para selecionar um conjunto pequeno das características que melhor separam as amostras positivas e negativas, visto que, embora cada filtro de Haar possa ser computado sobre as imagens integrais de maneira eficiente, calcular todo o conjunto de características seria uma tarefa computacional extremamente custosa e, portanto, inviável.

O algoritmo *AdaBoost* foi introduzido por Freund e Schapire (1995) e o seu funcionamento básico consiste em combinar de forma ponderada vários classificadores fracos até gerar um classificador forte. Conforme Gonzalez e Velásquez (2019), esta ideia está embasada na afirmação de que vários classificadores fracos, sendo estes ligeiramente melhores que escolhas aleatórias, podem combinar-se para formar um classificador de maior precisão, sempre que houver um número suficiente de amostras para o treinamento. O conceito por trás desse algoritmo é ilustrado pela Figura 14.

Figura 14 – Ilustração do funcionamento do algoritmo *AdaBoost*.



Fonte: Adaptado de Shafique e Hato (2014)

Para cada característica, o classificador fraco determina o limiar ótimo da função de classificação, de modo que o menor número possível de exemplos seja classificado

incorretamente (VIOLA; JONES, 2001). O classificador fraco $h_j(x)$ pode ser matematicamente descrito como:

$$h_j(x) = \begin{cases} 1, & \text{se } p_j f_j(x) < p_j \theta_j \\ 0, & \text{caso contrário} \end{cases} \quad (12)$$

Onde, para cada imagem x , normalmente de tamanho 24x24, o classificador fraco h_j consiste de uma característica de Haar f_j , um limiar θ_j , que determinará se a imagem x será classificada como positiva ou negativa e uma paridade p_j , indicando a direção da desigualdade. O *AdaBoost* pode ser especificado conforme o algoritmo 1.

Algoritmo 1: AdaBoost

Entrada: n imagens (x_i, y_i) , $y_i = 0$ para negativas e $y_i = 1$ para positivas.

Saída: classificador forte $H(x)$

```

1  início
2  * inicializar os pesos, sendo  $m$  e  $l$ , o número de negativos e positivos, respectivamente
3  para  $i \leftarrow 1$  até  $n$  faça:
4      se  $y_i = 0$  então:
5           $w_{1,i} \leftarrow \frac{1}{2m}$ 
6      senão
7           $w_{1,i} \leftarrow \frac{1}{2l}$ 
8      fim
9  fim
10 * T classificadores fracos
11 para  $t \leftarrow 1$  até  $T$  faça:
12     * normalizar os pesos, para que  $w_t$  seja uma distribuição de probabilidade
13      $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$ 
14     * selecionar o melhor classificador fraco com relação ao erro ponderado
15      $h_t(x) \leftarrow$  classificador fraco  $h_t$  com o menor erro  $\epsilon_t = \sum_i w_i |h_t(x_i) - y_i|$ .
16     * atualizar os pesos, sendo  $\beta_t = \frac{\epsilon_t}{1 - \epsilon_t}$ 
17     se  $x_i$  foi classificado corretamente então
18          $e_i \leftarrow 0$ ;
19     senão
20          $e_i \leftarrow 1$ ;
21     fim
22      $w_{t+1,i} \leftarrow w_{t,i} \beta_t^{1-e_i}$ 
23 fim
24 fim

```

Fonte: Adaptado de Viola e Jones (2001).

Ao finalizar, o classificador forte é:

$$H(x) = \begin{cases} 1, & \text{se } \sum_{t=1}^T \log\left(\frac{1}{\beta_t}\right) h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \log\left(\frac{1}{\beta_t}\right) \\ 0, & \text{caso contrário} \end{cases} \quad (13)$$

A principal vantagem do *AdaBoost* como mecanismo para seleção de características, está na velocidade de aprendizagem. Usando esse algoritmo, a complexidade computacional do aprendizado de um classificador equivale à $O(MNK)$, onde, M é a quantidade de classificadores fracos, N é o total de exemplos analisados e K diz respeito ao número de características. Em cada rodada, as dependências das características selecionadas anteriormente, são eficiente e compactamente codificadas utilizando os pesos dos exemplos. Esses pesos podem então ser empregues na avaliação de um determinado classificador em tempo constante (VIOLA; JONES, 2004).

2.4.4 Cascata de classificadores

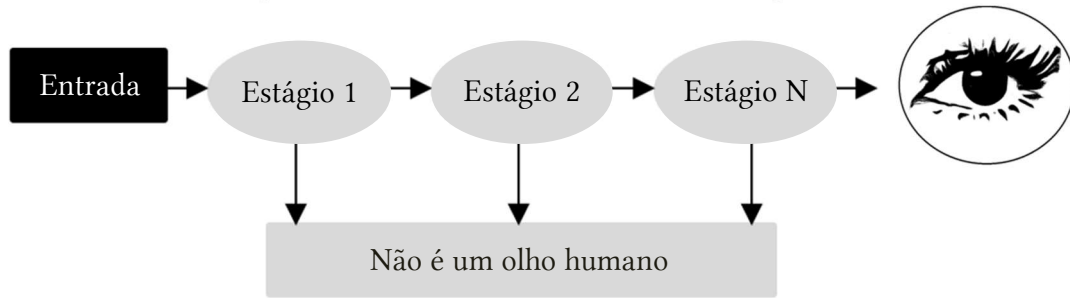
Na etapa final do algoritmo, é realizada a concatenação dos classificadores treinados com *AdaBoost*, em uma estrutura de cascata, também conhecida por cascata de atenção, com o objetivo de simplificar o treinamento e otimizar a detecção dos objetos. Através desse mecanismo, as regiões irrelevantes que não possuem características semelhantes a procurada, serão descartadas imediatamente, para então reservar os classificadores mais específicos para as regiões mais propensas a incluir o objeto de interesse (ZAGHETTO *et al.*, 2017).

Segundo Uddin e Akhi (2016), o sistema de detecção de objetos escaneia exaustivamente todas as sub-janelas de uma imagem, fazendo com que a análise dessas sub-janelas se torne intratável, quando o classificador forte selecionar milhares de características. Portanto, através de uma árvore de decisão degenerada para limitar o número de descritores computados para cada sub-janela, é possível contornar esse problema, reduzindo drasticamente o custo computacional.

O procedimento em cascata permite obter altas taxas de detecção, rejeitando o máximo possível de sub-imagens que não apresentam o objeto de interesse e ajustando a cada estágio, o limiar do classificador de modo que a taxa de falsos negativos fique próxima de 0. Dessa forma, um objeto somente será detectado com êxito, se a sub-janela ao qual pertence for aprovada por todos os classificadores da cascata (VIOLA; JONES, 2001). A Figura 15 exibe

uma representação esquemática de uma detecção em cascata com N estágios, que examina se a sub-janela de entrada contém um olho humano ou não.

Figura 15 – Cascata de classificadores com N estágios.



Fonte: Adaptado de Macedo e Gomes (2020).

Conforme Viola e Jones (2004), o processo de treinamento em cascata envolve uma série de metas e objetivos relacionados à detecção e desempenho, onde a seleção da quantidade de estágios na cascata e o tamanho de cada estágio deve ser o suficiente para alcançar uma taxa elevada de detecção, uma baixa ocorrência de falsos positivos e ainda reduzir o custo computacional. Na maioria dos casos, os classificadores que possuem mais características, obtêm índices maiores de detecção e taxas menores de falsos positivos, todavia, quanto maior for a quantidade de características, maior será o tempo de processamento demandado. Dado uma cascata de classificadores, a taxa de falsos positivos F é definida pela equação:

$$F = \prod_{i=1}^K f_i \quad (14)$$

Onde K representa o número de classificadores e f_i é a taxa de falsos positivos do i -ésimo classificador. A taxa de detecção D deste mesmo classificador é:

$$D = \prod_{i=1}^K d_i \quad (15)$$

Sendo K o número de classificadores e d_i a taxa de detecção do i -ésimo classificador. Após a definição das taxas de detecção e de falsos positivos, é possível determinar também, taxas específicas para cada estágio. O número de características avaliadas durante o escaneamento de uma imagem real é um processo probabilístico, tal que cada sub-janela irá percorrer a cascata, até que seja rotulada como negativa ou em circunstâncias raras, como positiva, caso aprovada por todos os classificadores.

Viola e Jones (2004) afirmam que o comportamento esperado deste processo é determinado pela distribuição das sub-janelas da imagem em um conjunto típico de teste. A

medida chave de cada classificador é a sua taxa de positivos, a proporção de sub-janelas rotuladas que potencialmente contém o objeto de interesse. O número esperado de características avaliadas N é dado pela equação:

$$N = n_0 + \sum_{i=1}^K \left(n_i \prod_{j<i} p_j \right) \quad (16)$$

Onde K diz respeito ao número de classificadores, p_j e n_i são a taxa de positivos e o número de característica do i -ésimo classificador, respectivamente. O algoritmo 2 apresenta o pseudocódigo proposto por Viola e Jones (2004), que permite construir uma eficiente cascata de classificadores.

Algoritmo 2: Algoritmo de treinamento do classificador em cascata

Entrada: f : taxa máxima de falsos positivos aceitável por estágio
 d : taxa mínima de detecção aceitável por estágio
 F_{target} : taxa de falsos positivos geral desejada
 P : conjunto de amostras positivas
 N : conjunto de amostras negativas

Saída: classificador treinado

```

1 início
2    $F_0 \leftarrow 1.0$ ;  $D_0 \leftarrow 1.0$ ;  $i \leftarrow 0$ ;
3   enquanto  $F_i > F_{target}$  faça:
4      $i \leftarrow i + 1$ ;
5      $n_i \leftarrow 0$ ;  $F_i \leftarrow F_{i-1}$ ;
6     enquanto  $F_i > f \times F_{i-1}$  faça:
7        $n_i \leftarrow n_i + 1$ ;
8       * Utilizar  $P$  e  $N$  para treinar o classificador com  $n_i$  características utilizando AdaBoost.
9       * Avaliar o classificador em cascata atual sobre o conjunto de validação para determinar
         $F_i$  e  $D_i$ .
10      * Reduzir o limiar para o  $i$ -ésimo classificador até que o classificador em cascata atual
        tenha uma taxa de detecção de pelo menos  $d \times D_{i-1}$  (isso também afeta  $F_i$ )
11    fim
12     $N \leftarrow \emptyset$ 
13    se  $F_i > F_{target}$  então
14      avaliar o detector em cascata atual no conjunto de imagens negativas e colocar qualquer
        falsa detecção no conjunto  $N$ 
15    fim
16  fim
17 fim

```

Fonte: Adaptado de Viola e Jones (2004).

3 TRABALHOS CORRELATOS

A detecção e o rastreamento do olhar tem sido um campo de pesquisa muito ativo nos últimos anos, sendo amplamente investigado como método alternativo de comunicação e interação, por ser considerado menos complicado do que outros procedimentos, como reconhecimento de voz ou sinais de eletroencefalograma (EEG) ou eletrocardiograma (ECG), e por adicionar conveniência à uma variedade de aplicações. Existem diferentes abordagens introduzidas nas pesquisas com foco no rastreamento ocular, a grande maioria envolve algoritmos baseados em visão computacional (AL-RAHAYFEH; FAEZIPOUR, 2013). A seguir são destacados quatro trabalhos relacionados à abordagem proposta, que impõem diversificadas metodologias de desenvolvimento.

Foi proposto por Nasor *et al.* (2018), um algoritmo implementado com o uso das ferramentas pertencentes ao software MATLAB®, que permite controlar o movimento do cursor da tela de um computador usando o posicionamento da íris. Ao extrair fotos consecutivas do usuário através da webcam, o programa foi projetado para processar esses *frames* individualmente em uma velocidade de processamento muito alta, e comparar o deslocamento da íris em cada *frame* em relação ao *frame* inicial. O *frame* passa por vários estágios de processamento antes dos olhos serem rastreados. Após obter a imagem processada, é efetuado o cálculo de deslocamento da íris, para então direcionar o cursor para um ponto específico da tela.

Para avaliar os resultados, foi desenvolvida uma interface gráfica do usuário (GUI), composta por nove caixas. Conforme Nasor *et al.* (2018), o movimento da íris foi rastreado com precisão e o cursor foi movido com sucesso para todas as nove caixas na GUI de teste. Entretanto, os movimentos para cima e para baixo não foram altamente consistentes. O algoritmo também permite que, ao piscar prolongadamente, o usuário possa abrir e fechar pastas, arquivos e aplicações. A ação de clique foi testada com sucesso por meio da mudança na cor das caixas.

No trabalho de Santos *et al.* (2015), foi apresentada uma tecnologia assistiva que permite que pessoas com movimentos limitados, causados por problemas cervicais ou cerebrais, possam se comunicar por meio de um computador, com o mínimo de intrusão possível. Este sistema possibilita replicar as funções de um mouse e um teclado virtual, por meio da detecção dos movimentos dos olhos e pálpebras dos usuários. O sistema foi desenvolvido baseando-se no uso de técnicas de visão computacional. O método *Viola-Jones* foi utilizado para obter a região dos olhos, e o algoritmo *Hough-Circles*, foi aplicado na

identificação do centro dos círculos formados pela íris do usuário. É a partir da localização do centro da íris, que o sistema pode determinar a direção do olhar.

Após os experimentos constatou-se que, mesmo não possuindo a mesma precisão de um mouse tradicional e não utilizando todas as funções do computador, os resultados obtidos pelo método proposto foram considerados satisfatórios, levando em consideração que o acesso às funcionalidades do sistema é realizado apenas por um conjunto restrito de movimentos. Santos *et al.* (2015) informam que, apesar do software possuir operações limitadas, o projeto apresenta contribuições relevantes, por não necessitar de equipamentos especiais ou marcadores para o rastreamento do movimento ocular.

Santos, Abrantes e Jorge (2017), descreveram a implementação de um sistema de rastreamento ocular com o uso de métricas faciais, baseado apenas em dispositivos não intrusivos, como a câmera *Intel RealSense*, para o desenvolvimento de uma interface humano-computador mais natural. Esta câmera possui informações de profundidade, cores e infravermelho, que tornam a extração das características faciais mais robusta e precisa, diminuindo os impactos causados pela luz ambiente, que é considerada uma das principais causas de erros no rastreamento dos olhos.

Por meio de técnicas de processamento de imagens, o sistema implementado é capaz de converter os elementos faciais no ponto de foco correspondente na tela do computador. Duas abordagens para mover o cursor foram realizadas, uma usando o olhar e a outra, o rastreamento do nariz. Santos, Abrantes e Jorge (2017), concretizaram o desenvolvimento do método proposto através das seguintes etapas:

1. Aquisição da imagem por meio da câmera de vídeo;
2. Processamento das informações para extração das características faciais requeridas, como a orientação da cabeça e a posição dos olhos, pupila e nariz. Nesta etapa foram utilizados os algoritmos *Haar Cascade* para a detecção da face e dos olhos, a transformada de *Hough* para encontrar o círculo da pupila e o filtro de *Kalman* para suavizar os movimentos dos olhos;
3. Com base nos pontos das características extraídas da face, o sistema estima a posição do cursor na tela;
4. Além do posicionamento do cursor, o sistema realiza algumas ações solicitadas pelo usuário, como clicar ou mover uma janela, através da detecção do piscar dos olhos e da abertura e fechamento da boca;

Durante os experimentos, Santos, Abrantes e Jorge (2017) descreveram que, com base em testes de usabilidade, a abordagem usando o movimento dos olhos não teve precisão

suficiente. O usuário pôde movimentar o cursor horizontalmente, mas verticalmente quase nenhum movimento foi percebido, já que o movimento da pupila é mínimo nessas circunstâncias. Por outro lado, o rastreamento do nariz obteve precisão suficiente para efetuar as tarefas com exatidão. Ações como arrastar, soltar ou clicar com o mouse para minimizar uma janela foram possíveis de serem executadas.

Ngo *et al.* (2016) apresentaram uma ferramenta desenvolvida com o suporte do *Image Processing Toolbox* do software MATLAB®, para auxiliar pessoas com deficiência motora, na operação do mouse do computador, por meio da verificação do movimento da íris e da aparência da esclera usando operadores morfológicos. O desenvolvimento do sistema foi efetivado em cinco estágios:

1. Inicialização e calibragem da captura das imagens, para corresponder as várias condições de iluminação.
2. Detecção da face através do algoritmo *Viola-Jones*.
3. Após a detecção da face, a imagem é dividida em quatro quadrantes, de forma que os olhos estejam localizados nos dois quadrantes superiores. O algoritmo *Viola-Jones* é aplicado novamente, no entanto para a detecção dos olhos nesses dois quadrantes.
4. Posterior a confirmação da detecção dos olhos, faz-se o reconhecimento do olhar fixo. A abertura dos olhos define o movimento na vertical, enquanto que os direcionamentos na horizontal foram estabelecidos após dividir verticalmente a sub-imagem dos olhos em duas partes A e B, e computar a proporção horizontal entre a quantidade de pixels brancos da esclera em ambos os lados, denotado por $h = A/B$, de forma que, $h \geq 1$ e $h < 1$, indicam o olhar para a esquerda e direita, respectivamente.
5. Por fim, este reconhecimento é aplicado no controle do mouse.

Após os experimentos, Ngo *et al.* (2016) constataram que o usuário consegue mover o cursor até a posição desejada na tela e clicar com precisão de 0,5cm. A taxa de *frames* de 10 FPS (*frames* por segundo) é suficiente para analisar e processar a imagem, deixando o vídeo suave em tempo real. Entretanto, foram destacados como desvantagens: problemas referentes às condições de baixa luminosidade; na etapa de inicialização e calibração, o usuário precisa localizar as informações da esclera manualmente, levando cerca de um a dois minutos para concluir este procedimento; e o sistema precisa ser calibrado para novos usuários.

4 METODOLOGIA

Com o propósito principal de desenvolver um processo de software de baixo custo computacional, apto a propiciar a interação com o computador através do rastreamento dos olhos, para a inclusão digital de deficientes motores, foram efetuados levantamentos bibliográficos relativos às técnicas de visão computacional, processamento digital de imagens e aprendizagem de máquina, que foram projetadas para a detecção, identificação e o reconhecimento de padrões.

Com base neste estudo, foi desenvolvido um programa que permite o manuseio das teclas direcionais (*Arrow Keys*), por meio da captura e análise dos movimentos dos olhos em tempo real, a partir da webcam do computador. A captura dos olhos foi efetuada através do método de aprendizagem de máquina *Haar Cascade* e o mapeamento dos movimentos oculares foi executado mediante a identificação da pigmentação branca pertencente a membrana externa do globo ocular, conhecida por esclerótica ou esclera.

O programa foi implementado através da linguagem de programação C++, com o suporte dos algoritmos otimizados de visão computacional e aprendizagem de máquina, pertencentes à biblioteca de código aberto OpenCV 4.1.0 (OPENCV, 2019). A biblioteca OpenCV já fornece em seu pacote de arquivos, classificadores pré-treinados que permitem inclusive, a detecção dos olhos. Todavia, com o intento de utilizar um classificador que apresentasse maiores taxas de acertos durante a detecção, foi realizado o treinamento de um novo classificador de haar, exclusivo para o sistema proposto.

O processo de treinamento foi executado baseando-se no trabalho de Rezaei (2014), onde foram utilizadas as aplicações: *objectmarker.exe*, *createsamples.exe*, *haartraining.exe* e *haarconv.exe*. As configurações do computador utilizado para o treinamento, estão descritas na Tabela 1. Os próximos subtópicos apontam as etapas seguidas para o desenvolvimento do programa, descrevendo os detalhes técnicos da implementação.

Tabela 1 – Configurações do computador.

Processador	Intel® Core™ i7-5500U CPU @ 2.40GHz
Tipo de Sistema	64 bits, processador com base em x64
Memória RAM	8,00 GB
Sistema Operacional	Windows 10

Fonte: Elaborada pela autora.

4.1 Amostras positivas e negativas

São necessários para o treinamento, um conjunto de amostras positivas que contém o objeto de interesse, neste contexto, o olho humano, e um conjunto de exemplos que não exibe o alvo que se deseja detectar (amostras negativas ou *background*). Conforme mencionado por Uddin e Akhi (2016), devido os grandes desafios do reconhecimento de objetos, ocasionados pelos inúmeros pontos de vista e ambientes em diferentes iluminações e contrastes, as imagens pertencentes às amostras positivas, devem ser escolhidas de modo que representem as mais variadas circunstâncias possíveis, em termos de posicionamentos, iluminância e assim por diante. Com fundamento neste princípio, foram utilizadas ao todo 13.117 imagens, onde 5117 possuem somente o olho humano e 8000 não apresentam o alvo de interesse.

As imagens positivas foram adquiridas através da internet e do banco de imagens *Appearance-based Gaze Estimation in the Wild (MPIIGaze)*, proposto por Zhang *et al.* (2015). Este conjunto de dados foi escolhido por ser, de acordo com os autores, significativamente mais variável do que os já existentes, em relação à aparência e iluminação. Além deste *dataset*, foi também utilizada algumas imagens inclusas na base de dados *Real and Fake Face Detection*, disponibilizado pelo laboratório de fotografia e inteligência computacional do departamento de Ciência da Computação da Universidade de Yonsei na Coreia do Sul, através da plataforma *Kaggle* (YONSEI UNIVERSITY, 2019). A Figura 16 exibe algumas das imagens positivas empregadas no treinamento do classificador.

Figura 16 – Imagens positivas contendo olhos humanos.



Fonte: Elaborada pela autora.

As amostras negativas são compostas de imagens arbitrárias oriundas da internet e na maior parte, imagens de regiões da face humana, adquiridas também através da internet e do conjunto de dados *Real and Fake Face Detection* da Universidade de Yonsei, de modo que o *background* estivesse o mais próximo possível do local em que os olhos serão encontrados. Ao inserir determinadas áreas faciais no conjunto de amostras negativas, é possível reduzir

consideravelmente a taxa de falsos positivos, que poderiam ser indicados na própria face. Em todas as imagens negativas que contém a face frontal, os olhos foram escondidos através de um quadrado branco. A Figura 17 é um exemplo das imagens negativas aplicadas no treinamento.

Figura 17 – Alguns exemplos do conjunto de imagens negativas.



Fonte: Elaborada pela autora.

4.2 Treinamento do classificador

O Haar Cascade é um algoritmo de aprendizagem supervisionado, portanto, faz-se necessário informar ao computador quais imagens são positivas e quais são negativas (UDDIN; AKHI, 2016). Para o conjunto de amostras negativas, basta inserir em um arquivo de texto, a localização de cada uma das imagens. Já para os exemplos positivos, é necessário informar em um arquivo de texto, além da localização das imagens, a quantidade de objetos por imagem, e para cada objeto, indicar a coordenada do canto superior esquerdo, a largura e a altura em pixels.

Dois procedimentos diferentes foram realizados para obter as informações relativas às amostras positivas. O primeiro foi através da aplicação *objectmarker.exe*, que permite desenhar um retângulo ao redor de cada alvo, gerando automaticamente após o processo, um arquivo de texto contendo todas as informações requeridas. O segundo procedimento foi efetuado em virtude da grande quantidade de exemplos utilizados, que acabaria tornando a tarefa de manualmente especificar as localizações dos objetos, algo bastante demorado.

Diante disso, para adquirir esses atributos de maneira automatizada e eficiente, foi implementado e executado um procedimento em C++ e OpenCV, que carrega todas as imagens positivas de um diretório específico, estas imagens já devem exibir somente a região dos olhos, fazendo com que a coordenada do canto superior esquerdo do objeto seja (0, 0), e em seguida, todos os atributos necessários são lidos e salvos no arquivo de texto.

A etapa seguinte consiste de criar um arquivo de vetores contendo as coordenadas de cada objeto em formato binário e redimensionada para um determinado tamanho (UDDIN; AKHI, 2016), neste caso foi especificado 24x24. Este arquivo pôde ser produzido através da

aplicação *createsamples.exe*. Ao término do preparo dos dados, foi efetuado o treinamento do classificador com a aplicação *haartraining.exe*, onde foi estabelecida a quantidade máxima de 15 estágios. Finalizado o treinamento, foi gerado para cada estágio N_i , um arquivo de texto contendo as informações do classificador específico desta camada. Na etapa final, todos os dados de cada estágio foram combinados em um único arquivo em formato XML, através da aplicação *haarconv.exe*, originando a chamada cascata de classificadores de haar. O período de treinamento durou aproximadamente 7 horas.

4.3 Detecção dos olhos

A detecção dos olhos foi efetuada através da classe *CascadeClassifier*, pertencente à biblioteca OpenCV. Ao instanciar a classe, foi carregado o arquivo XML adquirido após o treinamento do classificador, e então executado o método *DetectMultiScale*, que implementa a classificação em cascata, percorrendo a imagem de entrada em diferentes escalas, e retorna uma lista de retângulos com suas respectivas coordenadas, indicando os locais onde os objetos foram encontrados com sucesso (KAEHLER, BRADSKI, 2016).

Antes da definição dos valores para os parâmetros do método, foi realizado o pré-processamento de cada *frame*, por meio da conversão da imagem em formato RGB para a escala de cinza, e aplicação da função *equalizeHist* do OpenCV, que efetua a equalização do histograma através da distribuição dos valores de brilho. A webcam do computador foi configurada para capturar as imagens com uma resolução de 640x480 pixels.

Após o tratamento da imagem, foram especificados como argumentos do método, o valor 1.09 para o fator de escala (*scaleFactor*), que determina o quanto uma imagem será redimensionada em cada uma das escalas. O valor 20 para a quantidade mínima de vizinhos (*minNeighbors*), que objetiva evitar falsas detecções, pois, em conformidade com Kaehler e Bradski (2016), os locais reais do objeto em uma imagem tendem a receber vários acertos na mesma área, porque os pixels e as escalas circundantes frequentemente sinalizam a sua presença. Através deste parâmetro é possível controlar o número mínimo necessário dessas detecções, para indicar a existência do alvo de interesse. Conseqüentemente, para este caso em específico, somente será apontada a presença do olho humano em uma imagem, se houverem pelo menos 20 detecções sobrepostas. Por fim, através do parâmetro “tamanho mínimo” (*minSize*), foi estabelecido que somente as regiões de dimensões mínimas de 70x70 pixels, é que serão consideradas. Todos estes procedimentos ocorrem para cada *frame* obtido através da webcam.

4.4 Captura dos movimentos dos olhos

Concluído o processo de detecção dos olhos com o método *detectMultiscale*, foi realizada a análise dos movimentos oculares, levando em consideração a cor branca da esclera e as quatro direções de interesse: cima, baixo, esquerda e direita. Para efetivar esta análise, foram selecionadas regiões de interesse (ROIs), através das coordenadas dos retângulos retornados pela função de classificação, contendo apenas os alvos detectados. Logo após a obtenção das ROIs, com o propósito de facilitar a identificação das colorações e reduzir a influência da iluminação sobre o objeto, foi realizada a conversão das imagens do formato RGB para o HSV, por este sistema de cores fornecer uma separação natural das informações cromáticas e de intensidade, conforme apresentado na sessão 2.2.2.

Para identificar a coloração, foi empregue o método *InRange* pertencente ao OpenCV, que retorna uma imagem binária, onde os pixels de coloração branca caracterizam os locais onde foram encontradas a cor especificada por um determinado limite inferior e superior (KAEHLER, BRADSKI, 2016). Os valores que mais se adequaram para representar a tonalidade de interesse foram definidos experimentalmente. A Tabela 2 exhibe os intervalos selecionados para cada componente do sistema HSV.

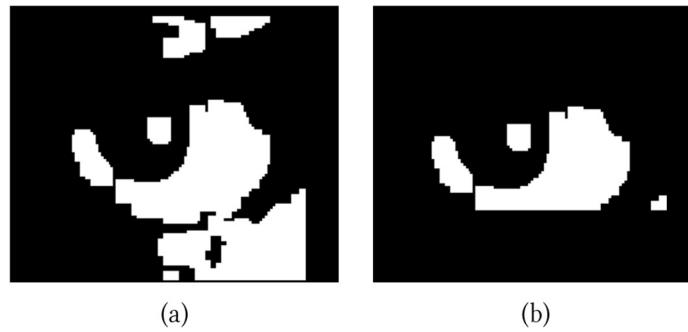
Tabela 2 – Limite inferior e superior da cor branca.

Componentes	Limites da coloração branca
H	60 até 180
S	20 até 255
V	150 até 255

Fonte: Elaborada pela autora.

Em sequência foi realizado o pós-processamento da imagem binária, por meio da aplicação das operações morfológicas de erosão e dilatação, cujos funcionamentos foram explanados na sessão 2.2.4. Ao final deste processo, notou-se que havia muitas regiões irrelevantes na imagem, que poderiam prejudicar a análise em decorrência da presença de ruídos. Para solucionar este problema, todos os pixels (i, j) da imagem binária *imb*, com valores equivalentes a 255, que estivessem localizados fora da região que vai de $(0, imb.h / 3.5)$ até $(imb.w - 1, imb.h - (imb.h / 3.5) - 1)$, foram alterados para a cor preta, tal que *imb.h* representa a altura da imagem e *imb.w* diz respeito à largura. É garantido que a região de interesse dos olhos se encontra normalmente dentro da área especificada acima. A Figura 18 é um exemplo do antes e o depois do processo de remoção das áreas irrelevantes.

Figura 18 – Antes e o depois da remoção das áreas irrelevantes.

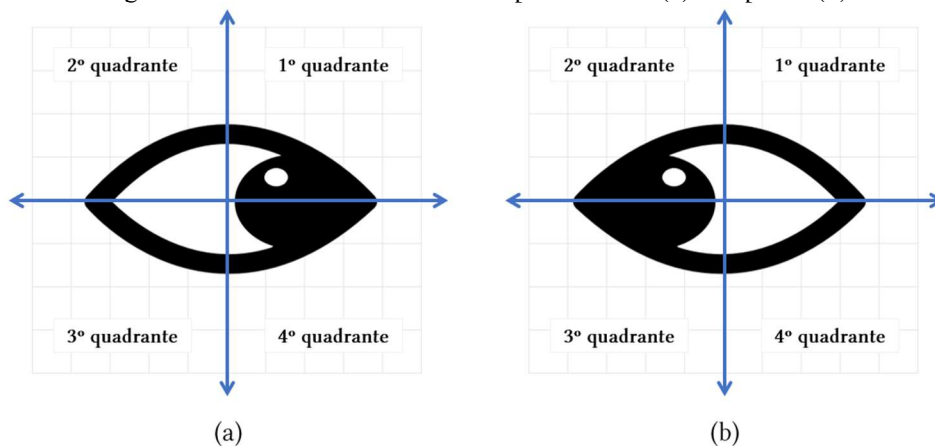


Fonte: Elaborada pela autora.

A análise da direção do olhar depende diretamente da identificação da coloração. Para garantir que a cor branca seja localizada com êxito na grande maioria das detecções, foi especificado que somente as ROIs que possuírem área mínima equivalente à 7000 pixels, é que serão analisadas. Isso se dá pelo motivo de que, quanto maior for a área do retângulo, mais próximo da câmera o usuário se encontra, conseqüentemente, mais fácil será a percepção da coloração. O valor mínimo da área foi definido após diversos experimentos.

O movimento na horizontal foi estabelecido através dos padrões elucidados pela Figura 19. Na letra (a), encontra-se o olhar para a direita, enquanto a letra (b) exhibe o olhar para a esquerda. Através da inclusão dos eixos vertical e horizontal, é possível notar que o movimento para à direita ocorre se a cor branca for predominante no 2° e 3° quadrante. Caso contrário, se a cor branca estiver predominantemente no 1° e 4° quadrante, o movimento diz respeito à esquerda. Caso a imagem não esteja espelhada, os quadrantes que definem as direções se invertem.

Figura 19 – Padrão dos olhos ao olhar para a direita (a) e esquerda (b).



Fonte: Elaborada pela autora.

O algoritmo percorre todas as ROIs detectadas por *frame*, o que normalmente são duas regiões relativas aos dois olhos, e armazena o acumulado da quantidade de pixels que representam a cor branca, nas variáveis que indicam as direções esquerda e direita,

considerando os seus respectivos quadrantes de referência. Após a obtenção do acumulado geral de cada um dos dois sentidos, é verificada qual das direções se encontra preponderante. A referência mínima de pixels para determinar uma direção, depende da área da imagem. A Tabela 3 exibe os valores mínimos aceitáveis para o movimento na horizontal, considerando a área das ROIs. Tais valores foram definidos após inúmeras análises sobre as imagens em diferentes condições e distâncias.

Tabela 3 – Quantidade mínima de pixels para determinar o movimento na horizontal.

Área da imagem	Referência mínima esquerda	Referência mínima direita
menor que 7500	180	160
entre 7500 e 12000	400	250
entre 12001 e 17000	700	600
maior que 17000	900	700

Fonte: Elaborada pela autora.

Com os valores de referência mínima estabelecidos, para averiguar se o usuário encontra-se olhando para à esquerda ou direita, foram estipuladas as seguintes condições: o olhar para à esquerda ocorre, se o acumulado geral da variável que representa à esquerda for maior ou igual a 600 pixels, maior que a variável referente à direita, e a diferença entre as duas variáveis for maior ou igual a 300.

Em algumas situações, ainda que o usuário esteja olhando para à esquerda, a variável em questão pode acabar não sendo superior à 600 pixels ou ao sentido oposto, em razão de problemas relacionados com a iluminação, ruídos ou distância. Para contornar problemas como esse, é verificado se a diferença entre as duas variáveis é maior ou igual à referência mínima para a esquerda. Quanto à direita, as condições são as mesmas, o que difere é a variável de interesse e o valor da referência mínima para a direção em estudo.

Quando objetiva-se definir a direção do olhar utilizando a abordagem de identificação da pigmentação da esclera, o movimento na vertical torna-se o mais difícil de se mapear, sobretudo ao examinar o olhar para baixo, que não exibe a coloração branca. Por consequência disso, depois de observar os padrões obtidos para às quatro direções, foi determinado que: se após quatro *frames* seguidos, a quantidade de pixels de cor branca nas variáveis que representam à esquerda e à direita for igual a 0, a probabilidade do usuário estar olhando para baixo é maior.

Contudo, faz-se necessário ainda, indicar a área mínima aceitável dos retângulos, uma vez que, quanto mais distante da câmera, mais difícil será a percepção da coloração, o que

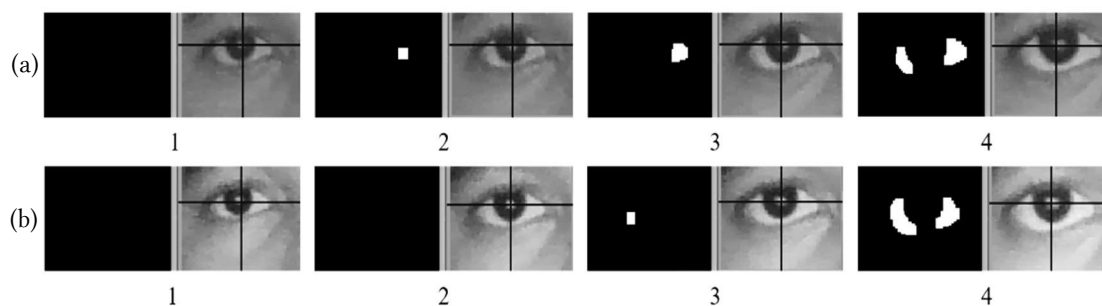
poderá levar o algoritmo a apontar erroneamente que o usuário encontra-se direcionando seu olhar para baixo. Apoiado neste princípio, determinou-se experimentalmente que, somente as sub-imagens com área maior que 9000 pixels, serão consideradas.

A Figura 20 é um exemplo do quanto a distância pode afetar a identificação da coloração branca. Esse problema é dependente também da iluminação ambiente. Nas letras (a) e (b), encontram-se as detecções em diferentes distâncias, nas condições de iluminação moderada (15 até 40 lux) e de baixa luminosidade (0 até 5 lux), respectivamente.

Essas densidades luminosas foram mensuradas com o auxílio do aplicativo Android “Lux Light Meter & Tools - Photometer PRO” instalado no dispositivo móvel Motorola One (XT1941-3), que consegue, através do sensor de iluminação ambiente do smartphone, capturar, com a maior precisão possível, os níveis de luminosidade presentes em um determinado local. A precisão da medição depende da qualidade do sensor de luz do dispositivo, por conseguinte, os dados podem diferir dos obtidos por diferentes aparelhos celulares e também por um luxímetro profissional.

Em oposição ao que é manifestado pelas Figuras a4 e b4, observa-se que, através das Figuras a1 e b1, ainda que o usuário esteja com o olhar voltado diretamente para a lente da câmera e não para baixo, o branco não é tão perceptível pelo algoritmo. Portanto, ao definir uma distância mínima aceitável quando se trata deste sentido em particular, é possível reduzir prováveis conflitos de direções.

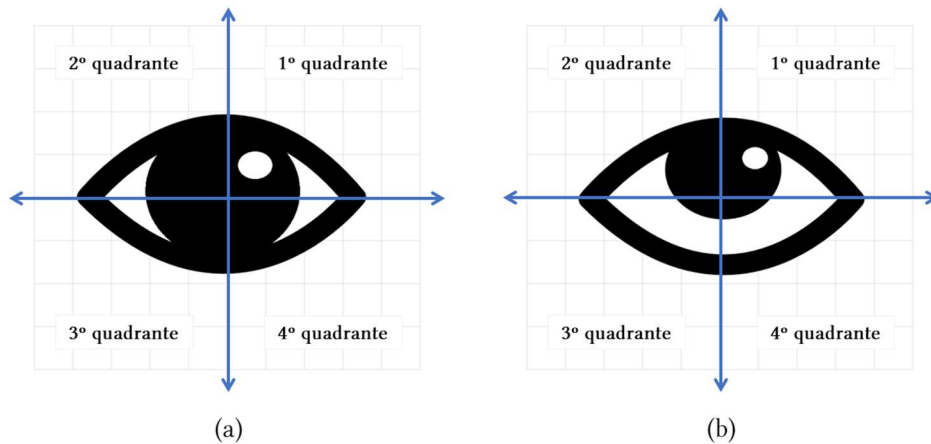
Figura 20 – Identificação do branco em diferentes distâncias, com moderada (a) e baixa (b) luminosidade.



Fonte: Elaborada pela autora.

Por último foi estabelecido o procedimento utilizado para capturar o olhar para cima. A Figura 21 exemplifica a característica padrão do movimento, onde as letras (a) e (b) apresentam o olhar direto para a lente da câmera e para cima, respectivamente. Através do exemplo (b), pôde-se constatar que, quando o olhar está direcionado para este sentido, ao contrário da Figura (a), uma região predominantemente branca abaixo do eixo horizontal fica exposta. Fundamentando-se neste princípio, foi possível efetuar o mapeamento desta direção.

Figura 21 – Padrão dos olhos ao olhar direto (a) e para cima (b).

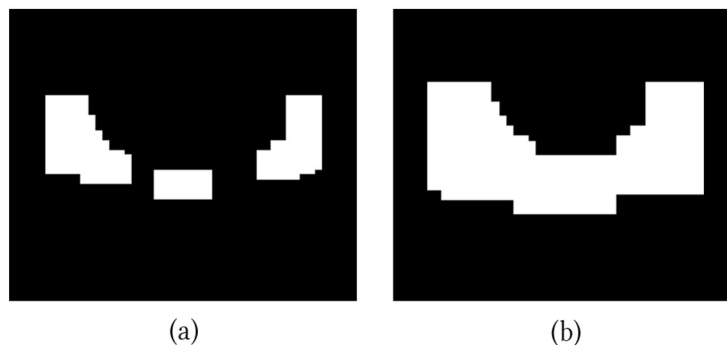


Fonte: Elaborada pela autora.

Para concluir o procedimento e evitar cálculos desnecessários, foi analisado somente a região dos olhos que contém a área de interesse. Esta região foi obtida através das coordenadas de um retângulo, cujos dois pontos (x, y) definidos empiricamente foram: $(img.w / 1.4, img.h / 3.9)$ até $(img.w - (img.w / 1.4), img.h)$, tal que $img.h$ e $img.w$ dizem respeito à altura e largura da imagem, respectivamente.

Dando prosseguimento, além de realizar os mesmos processos descritos anteriormente, referentes à identificação da coloração branca e ao pós-processamento da imagem binária, foi aplicada novamente a operação morfológica de dilatação, com o intuito de agrupar os componentes que pudessem ainda estar separados devido à presença de ruídos, já que para esta etapa, é primordial que a região branca abaixo do eixo horizontal esteja completamente unificada. A Figura 22a é um exemplo do problema que poderia acabar prejudicando a análise da direção, e a Figura 22b exibe a solução deste problema, mediante a aplicação do operador morfológico de dilatação.

Figura 22 – Antes e o depois da aplicação da operação de dilatação.



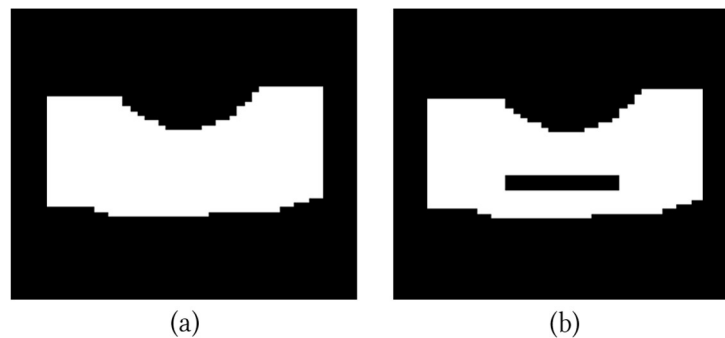
Fonte: Elaborada pela autora.

Com a conclusão de todos os pré e pós processamentos e a aquisição da imagem binária final, para indicar se o usuário encontra-se olhando para cima, é verificado se a cor

branca está presente em todos os pixels (i, j) que compõe a área que vai de $((imb.w / 2) - 8)$, $imb.h / 1.8$ até $((imb.w / 2) + 8, (imb.h / 1.7) - 1)$, de modo que, $imb.h$ e $imb.w$ indicam a altura e largura da imagem em pixels, respectivamente. O comprimento da região foi estipulado experimentalmente de forma que assegurasse um resultado satisfatório. A Figura 23 ilustra o resultado desta computação, onde a letra (a) exibe a detecção da cor branca presente na esclerótica, e a letra (b) indica a região que foi explorada para confirmar se o olhar está direcionado para cima.

Constata-se por meio da Figura 23a, que as variáveis que caracterizam à esquerda e à direita, irão apresentar uma grande quantidade de pixels de coloração branca. Com a condição que impõe que a diferença entre às duas variáveis deve ser maior ou igual a uma referência mínima específica, é possível evitar alguns dos conflitos que poderiam acontecer ao movimentar o olhar para cima.

Figura 23 – Área da imagem analisada para apontar se o olhar está direcionado para cima.



Fonte: Elaborada pela autora.

5 EXPERIMENTOS E RESULTADOS

Nos próximos subtópicos serão apresentados os resultados alcançados no processo de detecção dos olhos e do rastreamento do movimento ocular. Para efetuar os experimentos, foi utilizada a webcam do computador cujas configurações foram descritas no capítulo 3. Durante a execução do programa, para atingir resultados adequados, foi especificada que a dimensão dos *frames* deve ser de 640x480 pixels. Caso seja feita alterações na resolução da captura do vídeo, todos os valores definidos para o tamanho das áreas dos retângulos e para as referências mínimas da esquerda e direita, deverão ser modificados, de forma que a indicação das direções possa funcionar apropriadamente e sem conflitos no decurso da análise das imagens.

5.1 Métricas de avaliação

Com a finalidade de mensurar o desempenho do algoritmo de classificação proposto, foi calculada a matriz de confusão, que exibe uma comparação entre os dados reais e os dados previstos pelo detector. Conforme Ferreira (2018), as linhas da matriz correspondem às classes reais, enquanto que as colunas dizem respeito aos valores preditos pelo modelo. A Figura 24 retrata a organização desta matriz no que concerne problemas de classificação binária. Desta maneira originam-se os conceitos de:

- Verdadeiro Positivo (VP): ocorre quando o modelo detecta com exatidão os exemplos positivos, neste caso, o olho humano.
- Verdadeiro Negativo (VN): dar-se ao prever corretamente os dados negativos.
- Falso Positivo (FP): diz respeito aos dados negativos classificados equivocadamente como positivos;
- Falso Negativo (FN): erro produzido ao prever como negativas as amostras marcadas como positivas.

Para o contexto desta pesquisa, serão considerados somente os valores de VP, FN e FP. A partir desses dados é possível computar as métricas de avaliação conhecidas como: Precisão (P), Recall ou Sensibilidade (R) e F1 Score (F1). A taxa de VN não foi computada por ser considerada irrelevante para as circunstâncias do sistema. Para Maanna (2018), as métricas selecionadas para avaliar o modelo de aprendizagem de máquina são muito importantes. A escolha dessas métricas influencia na maneira como o desempenho dos algoritmos de aprendizagem são mensurados e comparados.

Figura 24 – Matriz de confusão para problemas de classificação binária.

		PREVISÃO DO CLASSIFICADOR	
		POSITIVO	NEGATIVO
VALORES REAIS	NEGATIVO	FALSO POSITIVO (FP)	VERDADEIRO NEGATIVO (VN)
	POSITIVO	VERDADEIRO POSITIVO (VP)	FALSO NEGATIVO (FN)

Fonte: Elaborada pela autora.

A métrica de precisão reflete a taxa de acertos do classificador ao prever a classe positiva (FERREIRA, 2018). Em conformidade com a fórmula 17, seu valor é definido como o número de verdadeiros positivos dividido pela soma entre o total de verdadeiros positivos e falsos positivos. Quanto maior for a precisão, menor será a taxa de falsas detecções.

$$P = \frac{VP}{VP + FP} \quad (17)$$

O Recall exibe a porcentagem de acertos que o classificador possui na classe positiva (FERREIRA, 2018), ou seja, o quão frequente é a detecção dos exemplos de uma determinada classe. Seu valor é obtido através da fórmula 18, onde o número de verdadeiros positivos é dividido pela taxa de verdadeiros positivos somado a quantidade de falsos negativos. Um baixo valor de recall significa uma alta quantidade de falsos negativos.

$$R = \frac{VP}{VP + FN} \quad (18)$$

O F1 Score é a média harmônica entre a precisão e o recall. Este método de avaliação busca o equilíbrio entre as duas métricas apresentadas anteriormente (FERREIRA, 2018). Quanto maior for o valor de F1, melhor será a qualidade geral do modelo de classificação. Seu valor é adquirido através da equação:

$$F1 = 2 * \frac{P * R}{P + R} \quad (19)$$

5.2 Resultado da detecção dos olhos

Para avaliar a performance do modelo de classificação dos olhos, foram utilizadas 1021 imagens estáticas coloridas que compõe o conjunto de dados *Real and Fake Face*

Detection da Universidade de Yonsei. Cada uma dessas imagens dispõe de um rosto frontal com características físicas diversificadas, em uma variedade de ângulos e condições ambientais e de luminosidade. Por conseguinte, foi mensurado o desempenho na captura de um total de 2042 olhos diferentes.

Os parâmetros do método *detectMultiscale* são primordiais para a classificação. Os argumentos devem ser minuciosamente estipulados com a finalidade de garantir a qualidade da detecção, considerando os distintos cenários e as inúmeras conjunturas em que as imagens podem ser adquiridas. Em tese, quanto maior for o valor especificado para a quantidade mínima de vizinhos, maior será a precisão do algoritmo, entretanto, ao aumentá-lo, a taxa de verdadeiros positivos pode reduzir substancialmente.

Além disso, a escolha do valor ideal para o fator de escala pode assegurar melhores resultados. Em princípio, um fator de escala maior, tornará o processo de detecção mais veloz, porém com a taxa de acertos reduzida. Em contrapartida, um fator de escala menor, provocará uma detecção mais lenta, porém com acurácia elevada. Durante os experimentos, pôde-se constatar que todos os argumentos especificados para os parâmetros do método, viabilizaram resultados satisfatórios, a despeito da presença de alguns falsos positivos e negativos.

A Tabela 4 evidencia uma comparação dos resultados alcançados na detecção dos olhos entre o classificador proposto nesta pesquisa e o modelo pré-treinado da biblioteca OpenCV. A segunda, terceira e quarta coluna indicam a quantidade de verdadeiros positivos (VP), falsos negativos (FN) e falsos positivos (FP), e as três últimas colunas apresentam os valores adquiridos ao computar as métricas de Precisão, Recall e F1 Score, respectivamente.

Tabela 4 – Comparação entre os classificadores proposto e pré-treinado do OpenCV.

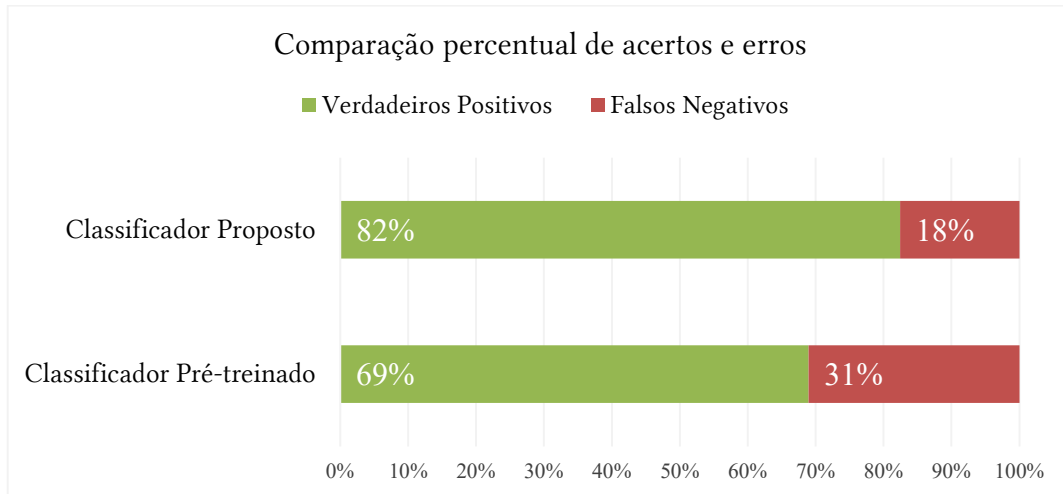
Classificador	VP	FN	FP	P	R	F1
Proposto	1683	359	244	0.87	0.82	0.84
Pré-treinado	1407	635	55	0.96	0.69	0.80

Fonte: Elaborada pela autora.

Embora o modelo pré-treinado do OpenCV tenha apresentado um índice bem menor de falsos positivos, observa-se que o classificador proposto demonstrou resultados superiores em relação às métricas de sensibilidade e F1 score. Para as circunstâncias desta pesquisa, a taxa de recall pode ser considerada a mais relevante, pois o objetivo principal é reduzir o número de falsos negativos, que em grande quantidade pode prejudicar a usabilidade do sistema. Logo, em função do problema conhecido como *Precision-recall tradeoff*, ao aumentar o valor de recall, a precisão do classificador acabará sendo afetada negativamente

(GÉRON, 2019). O gráfico da Figura 25 demonstra que, dentre os 2042 olhos do conjunto de teste, apenas 18% foram desconsiderados pelo classificador proposto, enquanto que o modelo pré-treinado do OpenCV ignorou 31%.

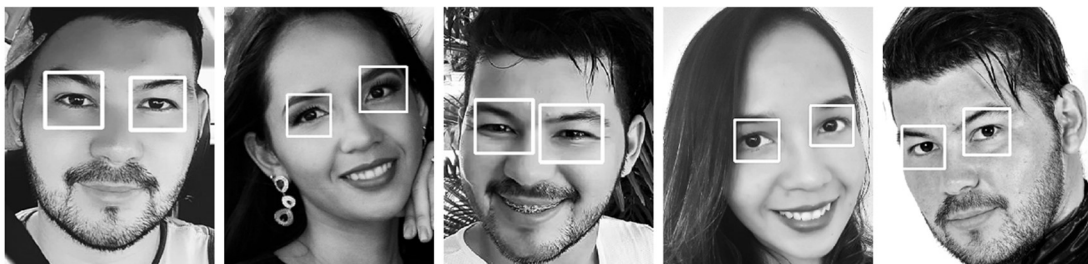
Figura 25 – Porcentagem de acertos e erros do classificador proposto e pré-treinado.



Fonte: Elaborada pela autora.

Através dos parâmetros impostos no sistema, é garantido que o usuário estará a uma distância adequada da câmera do computador, o que favorece a redução da porcentagem de falsos negativos e positivos durante a captura dos *frames* em tempo real. Além disso, o processo de identificação da coloração indiretamente previne que o algoritmo considere algumas das falsas detecções, devido à ausência da pigmentação branca em determinadas regiões da face que poderiam ser indicadas como falso positivo. A Figura 26 é um exemplo da detecção dos dois olhos na face humana, sem a presença de falsos positivos e negativos.

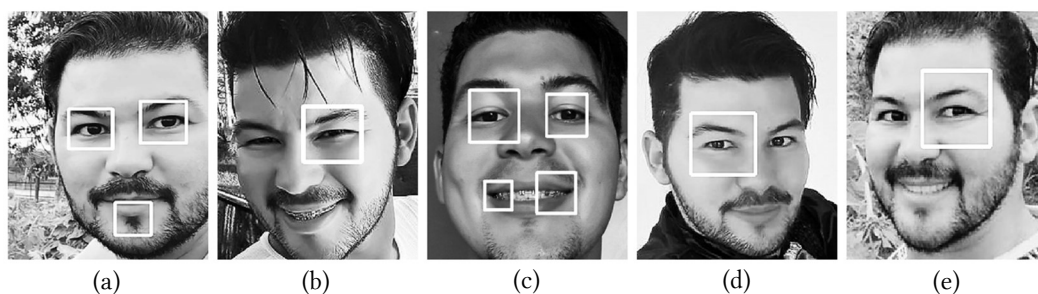
Figura 26 – Detecção dos olhos sem falsos positivos e negativos.



Fonte: Elaborada pela autora.

A Figura 27 demonstra alguns dos casos em que houve problemas relacionados à indicação de falsos positivos e negativos. De modo geral, a grande maioria das falsas detecções foram encontradas em regiões próximas aos lábios e nariz, conforme as imagens (a) e (c). Além disso, dependendo da resolução da imagem, do nível de abertura ocular, da inclinação e da rotação da cabeça para os lados, alguns dos olhos podem ser ignorados pelo classificador, como é o caso das faces (b), (d) e (e).

Figura 27 – Detecção dos olhos com a presença de alguns falsos positivos e negativos.



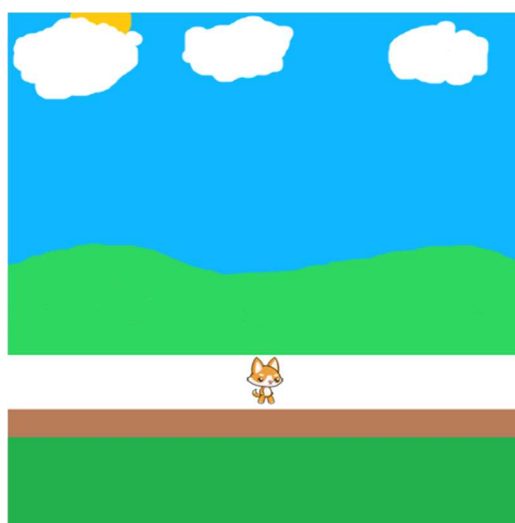
Fonte: Elaborada pela autora.

Uma das possíveis soluções para a redução desses problemas está na utilização de mais imagens para treinamento, o que demandará maior capacidade de processamento. Ademais, não foram efetuados treinamentos para a detecção dos olhos com a presença de óculos, diante disso, ao exercer experimentos nesta condição, o algoritmo poderá manifestar falhas na detecção, prejudicando a utilização do sistema.

5.3 Resultado do rastreamento dos olhos

Os experimentos referentes ao rastreamento dos olhos foram executados com o propósito de estimar a precisão e o tempo de resposta na detecção dos movimentos. Para efetuar esta avaliação, foi desenvolvida uma simples aplicação com dimensões de 500x500 pixels, que permite a movimentação do personagem¹ nas quatro direções de interesse. Ao especificar uma direção, o personagem terá um deslocamento de 20 pixels para o sentido indicado. A Figura 28 exibe a interface gráfica da aplicação.

Figura 28 – Aplicação desenvolvida para efetuar os experimentos no rastreamento dos olhos.

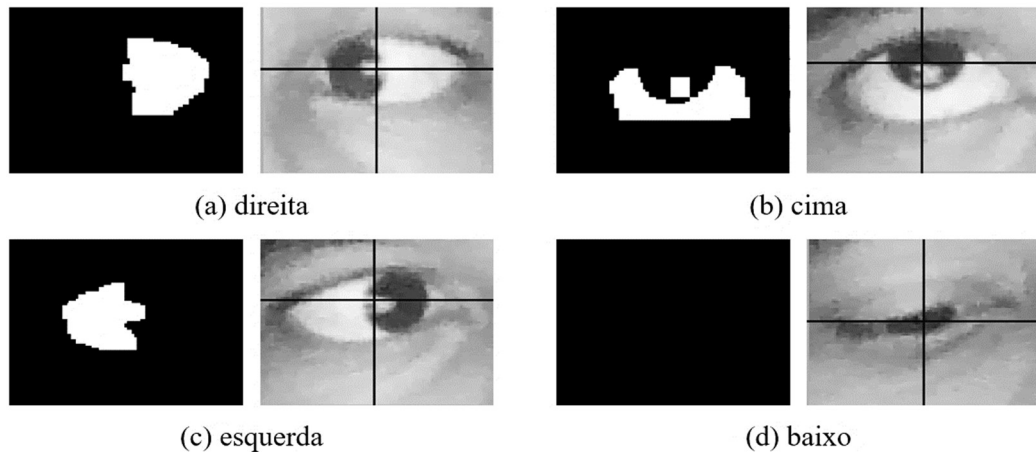


Fonte: Elaborada pela autora.

¹ Cats and dogs – Free Sprites. Disponível em: <https://www.gameart2d.com/cat-and-dog-free-sprites.html>. Acesso em: 6 de agosto de 2020.

Dependendo da intensidade da iluminação, a detecção da pigmentação branca pode ser afetada positiva ou negativamente. Observou-se que em lugares muito claros ou muito escuros, a presença de ruídos pode ser intensificada. Portanto, na busca de conquistar melhores resultados, faz-se necessário utilizar o sistema em ambientes com condições favoráveis e de fácil controle. A Figura 29 é um exemplo da identificação da coloração branca em tempo real, ao posicionar o olhar para as quatro direções de interesse desta pesquisa. Os *frames* foram obtidos no formato não espelhado, logo: a letra (a) refere-se à direita, a letra (b) diz respeito ao movimento para cima, e as letras (c) e (d) indicam os resultados para à esquerda e para baixo, respectivamente.

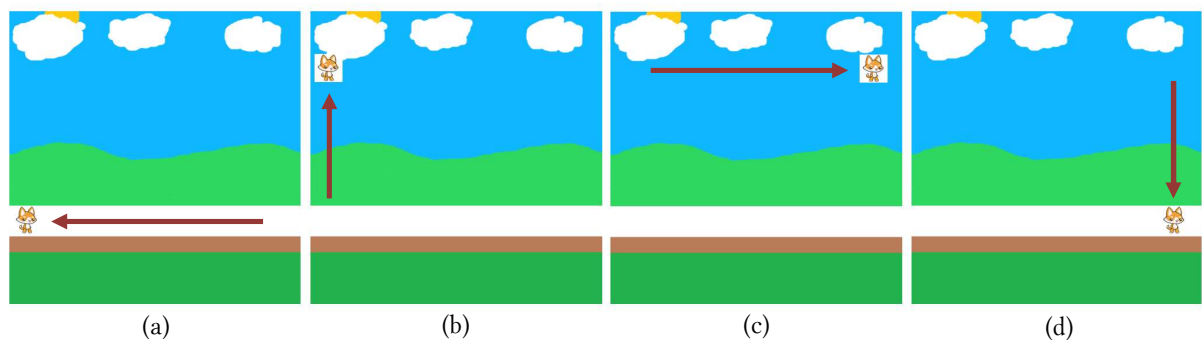
Figura 29 – Identificação da cor branca para as quatro direções.



Fonte: Elaborada pela autora.

A distância dos olhos até a lente da câmera, a claridade do espaço físico onde o sistema é manipulado, assim como a posição da cabeça, está essencialmente correlacionado com a precisão ao capturar a direção do olhar. Com o intento de simplificar o processo de avaliação, foram executadas uma série de experimentos que verificam o tempo demandado para direcionar o personagem da aplicação, levando em consideração a sequência de movimentos apresentadas pela Figura 30.

Figura 30 – Sequência de movimentos para avaliar o tempo de execução.



Fonte: Elaborada pela autora.

Com o auxílio de um cronômetro com precisão de 1/100 de segundos, foi previamente aferido o tempo de resposta individual das quatro direções, para em seguida apurá-las de forma conjunta. São necessários vários acionamentos das teclas direcionais para a conclusão de cada trajeto, a Tabela 5 exibe a quantidade mínima necessária para cada sentido.

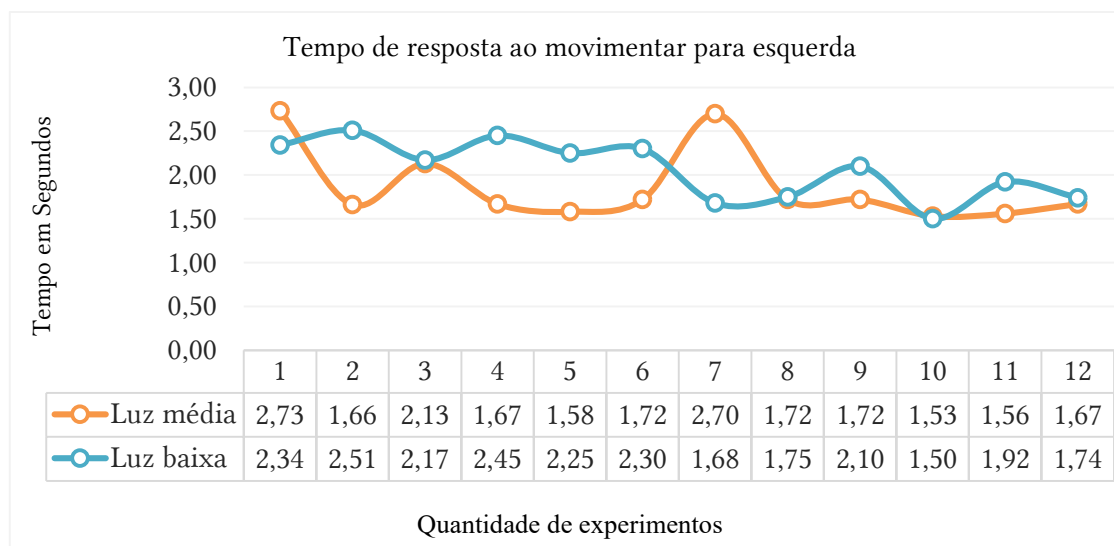
Tabela 5 – Número de acionamentos para a conclusão de cada percurso.

Sentido	Esquerda	Cima	Direita	Baixo
Número de acionamentos	21	13	21	13

Fonte: Elaborada pela autora.

Pretendendo avaliar o tempo de resposta em segundos de cada uma das quatro direções individualmente, foi realizado um experimento subdividido em duas sessões. Cada sessão foi composta de doze rodadas. A primeira se sucedeu em condições de baixa luminosidade (0 até 5 lux) e a segunda com intensidade luminosa equilibrada (15 até 40 lux). A motivação principal em dividir esta experimentação em duas etapas, está em demonstrar a influência da luminosidade sobre os resultados. No gráfico da Figura 31 são exibidas as durações necessárias para movimentar o personagem para à esquerda, em conformidade com a Figura 30a e a quantidade de acionamentos indicados na segunda coluna da Tabela 5.

Figura 31 – Tempo de resposta para movimentar o personagem para à esquerda.

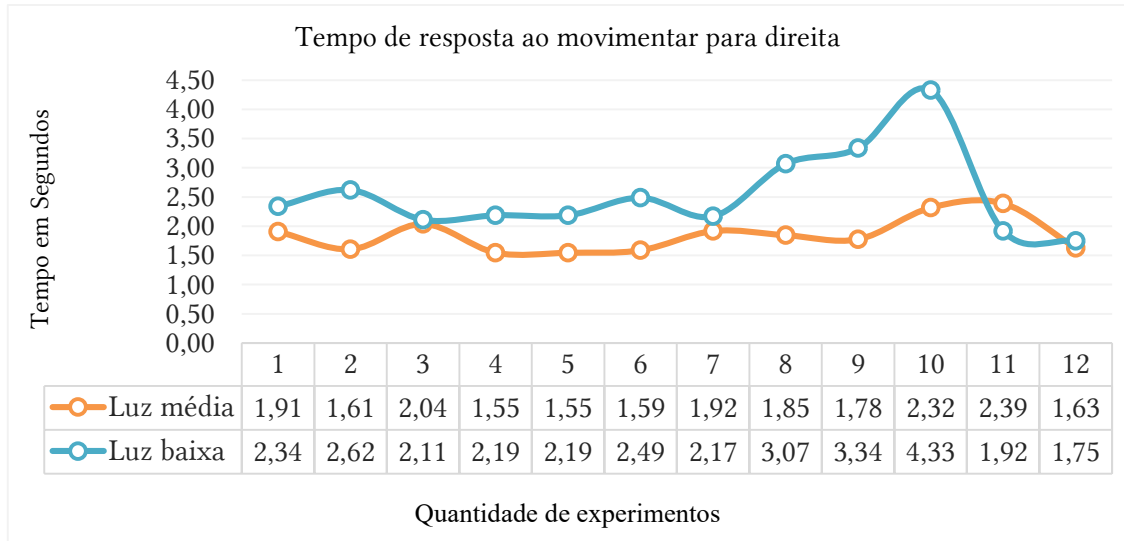


Fonte: Elaborada pela autora.

Observa-se através do gráfico da Figura 31, que as rodadas exercidas em iluminação moderada, obteve tempo médio equivalente a 1,87 segundos e foram completadas na sua maioria, com durabilidade bem menor que nos experimentos em condições de baixa luminosidade, que alcançou um tempo médio de 2,06 segundos. Parte desse resultado deve-se à presença intensificada de ruídos durante à baixa luminosidade, o que ligeiramente dificulta a detecção da direção de interesse. Apesar disso, ambos os experimentos demonstraram

resultados satisfatórios, principalmente quando considerado que esses foram os períodos para acionar pelo menos 21 vezes a direção de interesse. A Figura 32 apresenta o tempo demandado para direcionar o personagem para à direita, conforme a Figura 30c e a quantidade de acionamentos apontados na quarta coluna da Tabela 5.

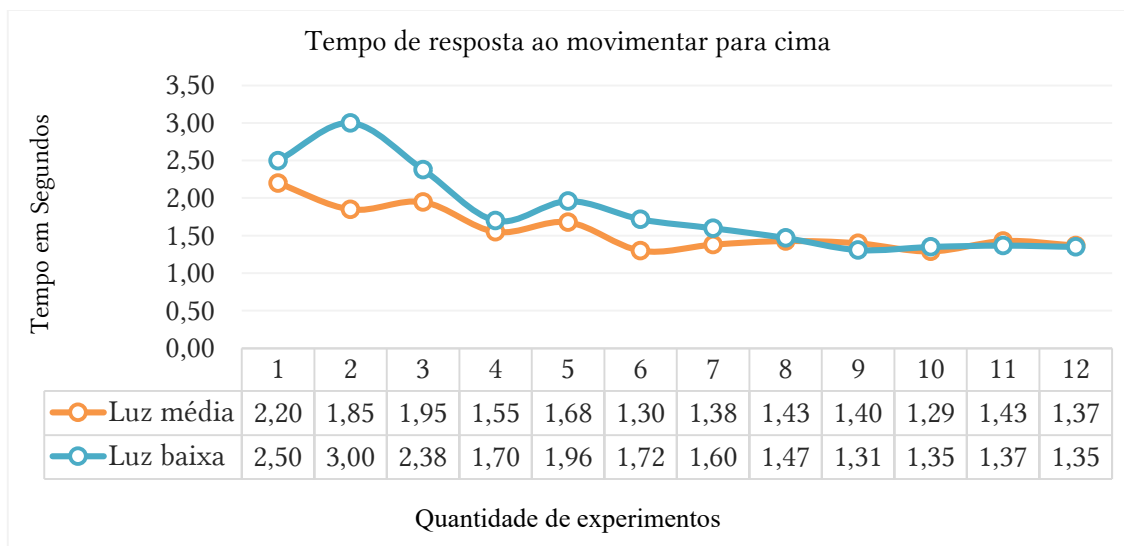
Figura 32 – Tempo de resposta para movimentar o personagem para à direita.



Fonte: Elaborada pela autora.

Os resultados evidenciados no gráfico da Figura 32 indicam características semelhantes aos experimentos referentes ao movimento para à esquerda, no que concerne à iluminação ambiente. Os tempos médios foram de 1,85 e 2,54 segundos, para as condições de média e baixa luminosidade, respectivamente. O gráfico da Figura 33 exibe o tempo demandado para movimentar o personagem da aplicação para cima, em concordância com a Figura 30b e a terceira coluna da Tabela 5.

Figura 33 – Tempo de resposta para movimentar o personagem para cima.

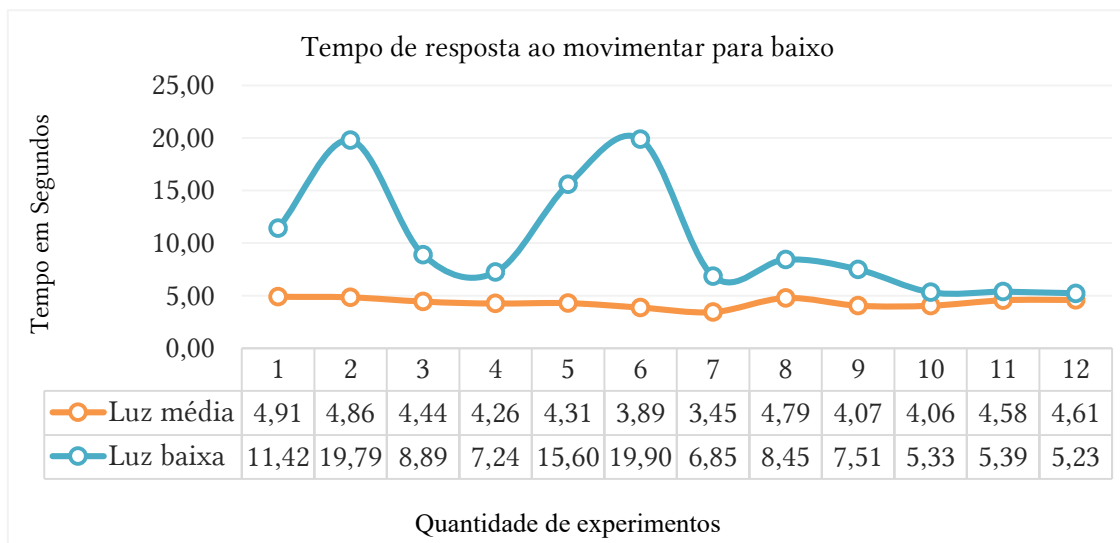


Fonte: Elaborada pela autora.

Constata-se através do gráfico da Figura 33, que o tempo demandado para movimentar o personagem para cima, sofreu uma ligeira redução na medida em que o número de experimentos foi aumentando. Conforme mencionado anteriormente, a posição da cabeça, a distância dos olhos para a lente da câmera, assim como outros fatores, influencia diretamente na detecção do movimento. Portanto, parte dessa diminuição foi provocada em decorrência da possibilidade do usuário se acostumar com o manuseio do sistema, de forma a encontrar os posicionamentos e ângulos que assegurem resultados mais eficientes.

O tempo médio para o contexto de iluminação moderada foi de 1,57 segundos, ao passo que para o cenário de baixa luminosidade, o tempo médio foi de 1,80 segundos. Esses períodos foram inferiores, se comparado aos tempos médios obtidos para os movimentos para à esquerda e direita, devido à quantidade de acionamentos mínimos demandados para concluir o experimento para esta direção específica, serem 13 ao invés de 21. A Figura 34 apresenta o tempo requerido para direcionar o personagem para baixo, em conformidade com a Figura 30d e a quantidade de acionamentos descritos na quinta coluna da Tabela 5.

Figura 34 – Tempo de resposta para movimentar o personagem para baixo.

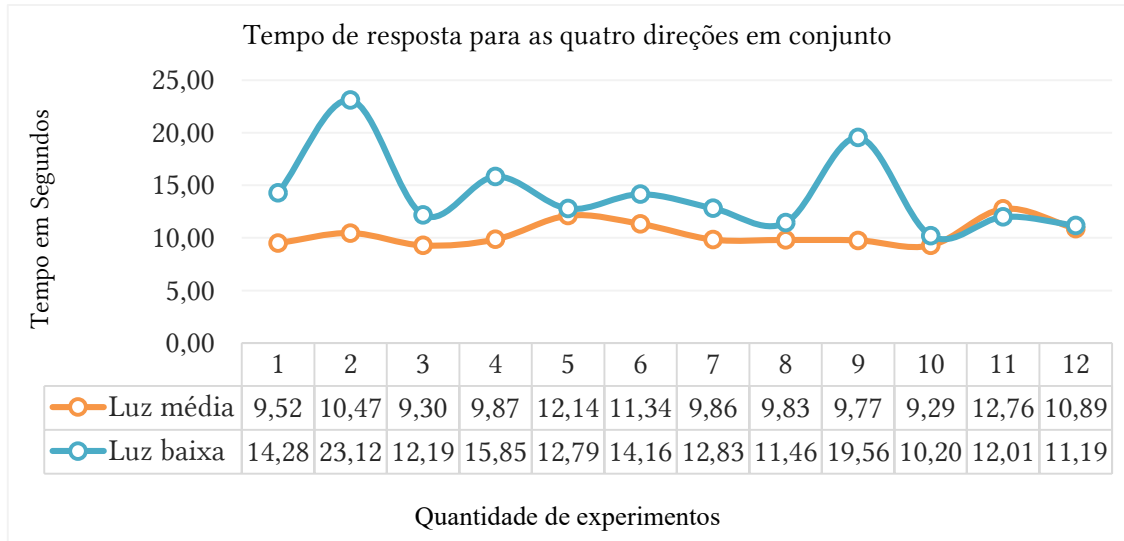


Fonte: Elaborada pela autora.

Dentre as quatro direções, o movimento para baixo foi o responsável por aumentar o tempo de processamento, por ser o mais difícil de detectar, principalmente por necessitar que o usuário fique próximo da lente da câmera e mantenha o olhar para baixo por um período prolongado, posto que o deslocamento para este sentido desenrola-se ao avaliar se as condições de um total de quatro *frames* consecutivos são favoráveis. Basta um dos alvos capturados indicar a existência da coloração branca, que a computação precisará ser refeita. Os tempos médios demandados para o usuário movimentar o personagem nas condições de média e baixa luminosidade foi de 4,35 e 10,13 segundos, respectivamente. Ao finalizar a

avaliação do tempo de resposta para cada sentido individualmente, foram efetuados os experimentos nas mesmas circunstâncias de iluminação, para as quatro direções em conjunto. O gráfico da Figura 35 demonstra os resultados alcançados no decorrer das rodadas, para movimentar o personagem conforme as instruções esclarecidas na Figura 30.

Figura 35 – Tempo de resposta para movimentar o personagem nas quatro direções em conjunto.



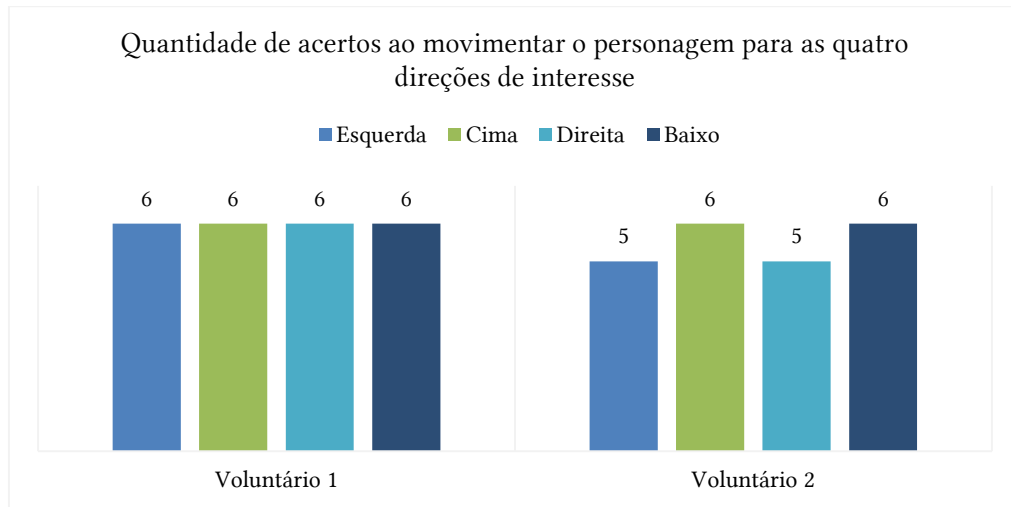
Fonte: Elaborada pela autora.

Os tempos médios para movimentar o personagem, considerando um total mínimo de 68 acionamentos, foram de 10,42 e 14,14 segundos, para os estados de luz média e luz baixa, respectivamente. Os resultados finais foram considerados satisfatórios, principalmente por se tratar de uma alta quantidade de acionamentos, o que evidencia que o sistema é capaz de extrair e analisar as informações oculares em tempo real, em uma velocidade de processamento suficiente para permitir o uso do sistema durante a realização de determinadas atividades computacionais, de maneira adequada e funcional.

Após a validação do tempo de processamento para as quatro direções de interesse, foram efetivados experimentos com a participação de dois voluntários, sem a medição do tempo de resposta, com o objetivo único de avaliar a eficácia do modelo proposto ao capturar os direcionamentos, considerando as distintas características físicas, que compreende vários aspectos como raça, formato, cor, textura e expressão. Testes com usuários distintos permitem avaliar se o sistema proposto poderá ser utilizado em maior escala.

As idades dos dois usuários selecionados para participar dos experimentos são de 6 e 55 anos. Cada participante realizou a experimentação seis vezes em condições de iluminação favorável. A Figura 36 indica a quantidade de acertos que cada voluntário possui ao movimentar o personagem em conformidade com as orientações da Figura 30 e a quantidade de acionamentos mínimos expostos na Tabela 5.

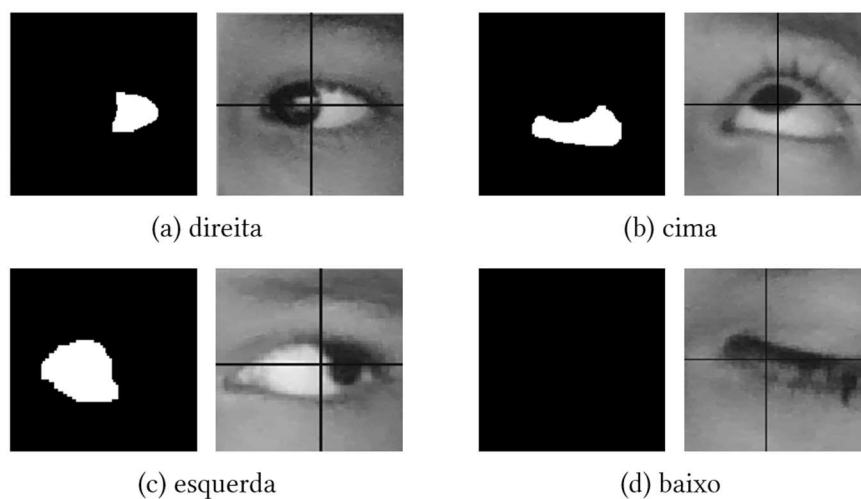
Figura 36 – Quantidade de acertos obtidos pelos dois usuários ao movimentar o personagem.



Fonte: Elaborada pela autora.

O primeiro voluntário foi capaz de movimentar com êxito, o personagem da aplicação para todas as quatro direções de interesse, ao longo das seis rodadas de avaliação. Os movimentos para à esquerda, para cima e para à direita, foram os que tiveram melhores retornos. Ainda que o movimento para baixo tenha sido finalizado com sucesso em todas as rodadas, observou-se que o usuário apresentou um ligeiro esforço para concretizar o direcionamento para este sentido em específico. Duas das razões que justificam a dificuldade do primeiro participante em direcionar o personagem para baixo, são as complicações relativas à distância apropriada até a lente da câmera, e ao posicionamento e ângulo adequado da cabeça, de forma que não prejudique a captura satisfatória dos olhos e da pigmentação. A Figura 37 é um exemplo da identificação da coloração branca presente na esclerótica do primeiro voluntário. Os *frames* não estão espelhados logo, as letras (a), (b), (c) e (d) referem-se aos movimentos para à direita, cima, esquerda e para baixo, respectivamente.

Figura 37 – Identificação da coloração branca presente na esclera do primeiro voluntário.

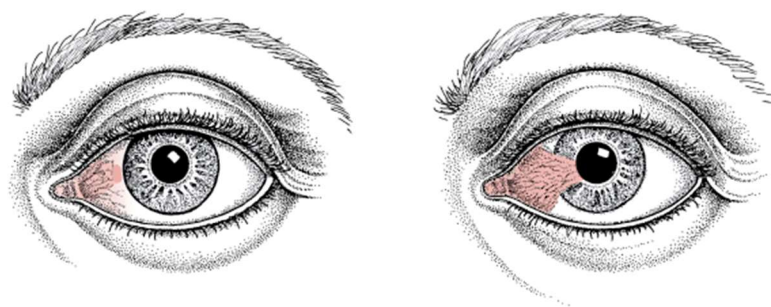


Fonte: Elaborada pela autora.

O segundo voluntário conseguiu movimentar o personagem da aplicação para cima e para baixo, em todas as seis rodadas da experimentação. Em apenas duas tentativas, uma para movimentar o personagem para à direita e outra para direcionar o personagem para à esquerda, o participante não obteve resultados satisfatórios. Os motivos principais para a ocorrência dessas duas falhas foram: dificuldades em posicionar convenientemente a cabeça; algumas imprecisões da aplicação em detectar a região ocular do usuário, o que também poderia, em algumas ocasiões, ser proveniente da primeira justificativa; e por último, existência de problemas associados à esclerótica que, por consequência, pode culminar em complicações para identificar a coloração branca em quantidade suficiente para determinar uma direção.

Doenças como pinguécua e pterídeo, dependendo do grau da lesão, podem dificultar a identificação da pigmentação branca presente na esclera pelo sistema proposto. Para o contexto do segundo voluntário, a existência do pterídeo em um dos olhos prejudicou a determinação de uma direção específica, pois o branco não foi identificado com exatidão. No entanto, constatou-se durante a experimentação que, dependendo da distância do usuário para a câmera e o posicionamento apropriado da cabeça, o problema pôde ser amenizado. Conforme Roat (2019), a pinguécua e o pterígio são formações carnosas na conjuntiva, que é a membrana que reveste as pálpebras e cobre o branco dos olhos. A Figura 38 é uma ilustração desses dois distúrbios oftalmológicos.

Figura 38 – Pinguécua à esquerda e pterídeo à direita.

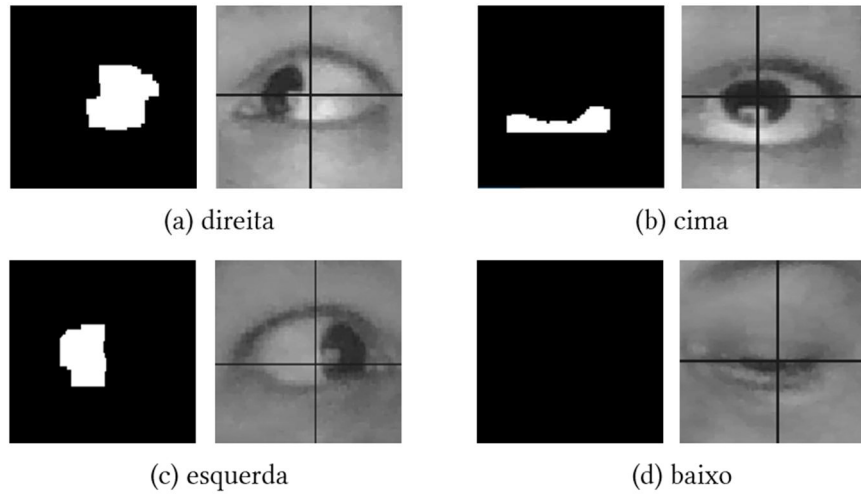


Fonte: Roat (2019).

Segundo Netto *et al.* (2006), a pinguécua é uma condição degenerativa extremamente comum em adultos, e se apresenta como uma pápula amarelada em ambos os lados da córnea, sendo mais frequente no lado nasal. Já o pterídeo consiste em uma lesão triangular semelhante à pinguécua, que invade a córnea, normalmente unilateral e situada no lado nasal. É reconhecida como uma doença de origem irritativa, devido à luz ultravioleta, o ressecamento ocular e o vento, sendo comum em pessoas que se expõem a estes fatores. A Figura 39 exhibe os resultados referentes à identificação da coloração branca presente na esclera do segundo

voluntário. Os *frames* não estão espelhados, portanto, as letras (a) e (b) indicam o olhar projetado para à direita e para cima, respectivamente, a letra (c) refere-se ao direcionamento para à esquerda, e a letra (d) retrata o resultado obtido ao projetar o olhar para baixo.

Figura 39 – Identificação da coloração branca presente na esclera do segundo voluntário.



Fonte: Elaborada pela autora.

6 CONSIDERAÇÕES FINAIS

Em concordância com os procedimentos metodológicos e os objetivos destacados neste projeto de pesquisa, foi proposto o desenvolvimento de um processo de software de baixo custo computacional e não intrusivo, capaz de efetuar a detecção e o rastreamento dos olhos, mediante algoritmos de visão computacional, processamento digital de imagens e aprendizagem de máquina. O sistema foi implementado com o uso da linguagem de programação C++, em conjunto com a biblioteca de código aberto OpenCV. O modelo proposto tem por objetivo principal auxiliar pessoas com deficiência motora, na execução de tarefas computacionais que necessitam do movimento de um cursor para uma direção específica, através do movimento dos olhos.

A metodologia de desenvolvimento desta pesquisa seguiu os princípios do método VOG e baseou-se no treinamento e aplicação da abordagem de aprendizagem de máquina *Haar Cascade* para efetuar a detecção das regiões oculares, e no uso de procedimentos específicos para identificar a pigmentação branca presente na esclerótica do globo ocular. Após a identificação da coloração, foram impostos parâmetros e condições que permitiram determinar o movimento dos olhos para quatro direções de interesse: cima, baixo, esquerda e direita. Ao finalizar, as ações processadas são utilizadas para o controle das teclas direcionais (*Arrow Keys*), que podem ser empregadas em inúmeras aplicações como jogos eletrônicos e sistemas de realidade virtual.

O detector dos olhos obteve desempenho geral satisfatório, alcançando uma taxa de verdadeiros positivos equivalente à (82%) sobre o conjunto de validação. Conseqüentemente, a taxa de falsos negativos foi de (18%). Já as métricas de precisão, recall e F1 score, tiveram valores correspondentes à 0,87, 0,82 e 0,84, respectivamente. Não foram efetuados treinamentos para a detecção dos olhos com o uso de óculos, portanto, ao efetuar experimentos nessas condições, o algoritmo poderá apresentar falhas. Além disso, é primordial que o número de falsos negativos seja reduzido ao máximo, dado que falhas na detecção dos olhos pode prejudicar o uso eficiente do sistema. Durante os experimentos, observou-se que dependendo do nível de abertura ocular e da inclinação e rotação da cabeça para os lados, alguns dos olhos podem ser ignorados pelo classificador. À vista disso, com o propósito de obter resultados com maior precisão, é primordial que o usuário posicione a face de maneira apropriada.

Na etapa de experimentação referente ao rastreamento dos olhos, foram conquistados resultados com potencial suficiente para permitir que o usuário utilize o sistema de forma

adequada e funcional. Constatou-se que o modelo proposto consegue detectar as quatro direções de interesse com êxito e com tempo de processamento apropriado para aplicações em tempo real. Observou-se que a intensidade da iluminação ambiente pode afetar a resposta do sistema, positiva ou negativamente. Usar o sistema em locais com condições de iluminação moderada e de fácil controle permite obter maiores taxas de acertos. Por outro lado, em ambientes com iluminação muito forte ou muito fraca, a identificação da coloração pode comprometer a precisão do sistema, levando à indicação incorreta das direções de interesse ou até mesmo, à não identificação das direções. Outros fatores que afetam diretamente os resultados da detecção do movimento ocular são: a distância dos olhos até a lente da câmera, o posicionamento e a rotação da cabeça.

O tempo de processamento para detectar os movimentos para cima, para à esquerda e para à direita foram aproximados. Em relação a captura do direcionamento para baixo, o tempo de resposta foi maior em virtude da necessidade de que o usuário projete o olhar para baixo por um período prolongado e encontre o posicionamento e a distância adequada para a lente da câmera. No experimento envolvendo a participação de dois voluntários, foi possível avaliar a robustez do modelo proposto diante de características físicas distintas. Ambos os participantes foram capazes de mover o personagem da aplicação de teste para as quatro direções de interesse. No entanto, durante a avaliação de um dos usuários, constatou-se que a existência de problemas que acometem a esclera, como pinguécua e pterídeo, podem impedir o reconhecimento da coloração branca em quantidade suficiente, prejudicando o uso efetivo do sistema.

Pretende-se para trabalhos futuros efetuar experimentos com mais voluntários, principalmente pessoas com deficiência motora, para obter uma avaliação mais detalhada da usabilidade e dos benefícios propostos pelo sistema. Além disso, com o objetivo de aprimorar ainda mais o desempenho do modelo proposto, está prevista a realização de maiores pesquisas relacionadas às técnicas de visão computacional e aprendizagem de máquina, de modo a proporcionar maior liberdade e precisão durante o processo de detecção dos olhos e do movimento ocular, considerando as inúmeras características físicas e as diversas condições ambientais em que o sistema pode ser manuseado. Para a continuidade do projeto, almeja-se também implementar novas funcionalidades, como a detecção de cliques e aplicação das ações processadas no movimento do cursor do mouse, para possibilitar aos usuários o aproveitamento máximo das funções computacionais, garantindo assim uma maior integração digital.

REFERÊNCIAS

- AL-RAHAYFEH, Amer; FAEZIPOUR, Miad. Eye tracking and head movement detection: A state-of-art survey. *In: IEEE journal of translational engineering in health and medicine*, v. 1, p. 2100212-2100212, 2013.
- ALLING, Alexander; POWERS, Nathaniel R.; SOYATA, Tolga. Face recognition: A tutorial on computational aspects. *In: Emerging Research Surrounding Power Consumption and Performance Issues in Utility Computing*. IGI Global, 2016. p. 405-425.
- ALVES, Dorirley Rodrigo. **Avaliação dos modelos de cores RGB e HSV na segmentação de curvas de nível em Cartas topográficas coloridas**. 2010. 67 f. Dissertação (Mestrado em Engenharia Elétrica) – Pontifícia Universidade Católica de Minas Gerais, Belo Horizonte, 2010.
- AZORÍN, José María *et al.* **A Interação de Pessoas com Deficiência com o Computador**. 2014.
- BARROS, Mayara Benício S.; SOUZA, Walla Nascimento. Inclusão Digital: um mapeamento sistemático de ferramentas e aplicativos para pessoas com deficiência. **RENOTE**, v. 18, n. 1, 2020.
- BELUSSI, Luiz; HIRATA, Nina. Fast QR code detection in arbitrarily acquired images. *In: 2011 24th SIBGRAPI Conference on Graphics, Patterns and Images*. IEEE, 2011. p. 281-288.
- BIGA, Lindsay M. *et al.* Anatomy & physiology. **Corvallis, Oregon: Oregon State University**. 2019. Disponível em: <https://open.oregonstate.edu/education/aandp/chapter/15-5-vision/>. Acesso em: 25 jul. 2020.
- BRASIL, Câmara dos Deputados. Lei nº 13.146, de 6 de julho de 2015. Institui a Lei Brasileira de Inclusão da Pessoa com Deficiência (Estatuto da Pessoa com Deficiência). **Diário Oficial da União**, 2015.
- BURGER, Wilhelm; BURGE, Mark J. Principles of digital image processing: Fundamental Techniques. **Springer Science & Business Media**, 2009.
- CAPELINI, Luiz Fernando. **O funcionamento do olho humano: uma sequência didática para o ensino de óptica geométrica no ensino médio**. 2019. 124 f. Dissertação (Mestrado em Ensino de Física) – Universidade Tecnológica Federal do Paraná, Campo Mourão, 2019.
- CARVALHO, Daniel Rodrigues. **Reconhecimento e Rastreamento da Direção do Olhar Para Auxílio de Deficientes Físicos**. 2015. 55 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Computação) – Universidade Federal da Bahia, Brasil, 2015.
- CARVALHO, José Oscar Fontanini de. O papel da interação humano-computador na inclusão digital. **Transinformação**, v. 15, n. 3, 2012.
- CAZELOTO, Edilson. Inclusão digital: uma visão crítica. **Editores Senac São Paulo**, 2019.

CHIBA, Elaine Ayumi. **Corte em grafos e segmentação de imagens utilizando um algoritmo aglomerativo de agrupamento hierárquico**. 2014. 52 f. Dissertação (Mestrado em Tecnologia) – Faculdade de Tecnologia, Universidade Estadual de Campinas, Limeira, 2014.

CORCOVIA, Lucas Oukus; ALVES, Renato Santos. APRENDIZAGEM DE MÁQUINA E MINERAÇÃO DE DADOS. **Revista Interface Tecnológica**, v. 16, n. 1, p. 90-101, 2019.

FERREIRA, Rodrigo Peres. **Reconhecimento de cenários baseado nas localizações dos fornecedores do governo federal**. 2018. 52 f. Dissertação (Mestrado Profissional em Computação Aplicada) – Universidade de Brasília, Brasília, 2018.

FRANÇA, Carlos Roberto; BORGES, José Antonio; SAMPAIO, Fábio Ferrentini. Tupi–Recursos de acessibilidade para educação especial e inclusiva dos deficientes motores. *In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, v. 1, n. 1, p. 591-600, 2005.

FREUND, Yoav; SCHAPIRE, Robert E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *In: European conference on computational learning theory: Eurocolt'95*, 1995. p. 23-37.

GÉRON, Aurélien. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. 2. ed. “**O'Reilly Media, Inc.**”, 2019.

GONZALEZ, Hirvin; VELÁSQUEZ, Sergio. Reconocimiento facial utilizando Viola-Jones y patrones binários. **Universidad Ciencia y Tecnología**, v. 23, n. 92, p. 57-63, 2019.

GONZALEZ, Rafael C.; WOODS, Richard C. Processamento digital de imagens. 3. ed. **Pearson Educación**, 2009.

IBGE. **Pesquisa nacional de saúde 2013: ciclos de vida: Brasil e grandes regiões**. 2015. Rio de Janeiro. Disponível em: <https://biblioteca.ibge.gov.br/visualizacao/livros/liv94522.pdf>. Acesso em: 19 ago. 2020.

KAEHLER, Adrian; BRADSKI, Gary. Learning OpenCV 3: computer vision in C++ with the OpenCV library. “**O'Reilly Media, Inc.**”, 2016.

LIENHART, Rainer; MAYDT, Jochen. An extended set of haar-like features for rapid object detection. *In: Proceedings. international conference on image processing*. IEEE, 2002. p. I-I.

MAANNA, Noura. **Image Classification using Machine Learning Techniques**. 2018. 65 f. Tesi di Laurea Magistrale (Mestrado em Engenharia Matemática) – Facoltà di Ingegneria, Politecnico di Torino, Torino, Itália, 2018.

MACEDO, Anne Livia da Fonseca; GOMES, Igor Ruiz. Algoritmo Haar Cascade Aplicado na Detecção das Placas de Parada Obrigatória e de Velocidade Máxima Permitida. **Anais do XI Computer on the Beach**, v. 11, n.1, p. 440-446, 2020.

MACHADO, Vinicius Ponte. **Inteligência Artificial. Universidade Estadual do Ceará.** 2017.

MAIA, Núbia Cristina de Freitas. **Fundamentos básicos da oftalmologia e suas aplicações. Palmas: EDUFT, 2019.**

MANZI, Filipe Augusto. **Aplicação de visão computacional para extração de características em imagens do olho humano.** 2007. 47 f. Trabalho de Conclusão de Curso (Engenharia da Computação com ênfase em Sistemas Embarcados) – Universidade de São Paulo, São Carlos, 2007.

MARQUES FILHO, Ogê; VIEIRA NETO, Hugo. **Processamento digital de imagens. Brasport, 1999.**

MAURI, César *et al.* Computer vision interaction for people with severe movement restrictions. **Human Technology: An Interdisciplinary Journal on Humans in ICT Environments**, v. 2, n. 1, p. 38-54, 2006.

MENDONÇA, André H. L. **Visão computacional no auxílio de deficientes visuais.** 2019. Trabalho de Conclusão de Curso – Departamento de Computação, Universidade Federal Rural do Semi-Árido, Mossoró, 2019.

MOHAMMED, Mohssen; KHAN, Muhammad Badruddin; BASHIER, Eihab Bashier Mohammed. **Machine learning: algorithms and applications. CRC Press, 2016.**

MONARD, Maria Carolina; BARANAUSKAS, José Augusto. Conceitos sobre Aprendizado de Máquina. *In: REZENDE, S. O. Sistemas inteligentes: fundamentos e aplicações.* Barueri: Manole, v. 1, n. 1, p. 89-114, 2003.

MONTEIRO, Gonçalo; PEIXOTO, Paulo; NUNES, Urbano. Vision-based pedestrian detection using Haar-like features. *In: Robotica 2006 - Scientific meeting of the 6th Robotics Portuguese Festival.* Portugal, 2006.

MOREIRA, Gustavo Costa Gomes. **Reconhecedor de objetos em vídeos digitais para aplicações interativas.** 2008. 70 f. Dissertação (Mestrado em Informática) – Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2008.

NASOR, Mohamed *et al.* Eye-controlled mouse cursor for physically disabled individual. *In: 2018 Advances in Science and Engineering Technology International Conferences (ASET).* IEEE, Abu Dhabi, 2018. p. 1-4.

NETTO, Augusto Adam *et al.* Prevalência de doenças conjuntivais no serviço emergencial de oftalmologia do Hospital Universitário da Universidade Federal de Santa Catarina. **Arquivos Catarinenses de Medicina**, v. 35, n. 4, p. 44-49, 2006.

NGO, Nguyen Hoai *et al.* Implement a Computer Mouse Control for Quadriplegic Disability Using Camera. *In: International Conference on Advanced Engineering Theory and Applications.* Springer, Cham, 2016. p. 278-288.

OMS, Organização Mundial da Saúde. **World report on disability 2011**. 2011. Disponível em: https://www.who.int/disabilities/world_report/2011/report.pdf?ua=1. Acesso em: 20 ago. 2020.

OPENCV. **Releases: OpenCV 4.1.0**. 2019. Disponível em: <https://opencv.org/releases/page/2/>. Acesso em: 03 jul. 2020.

PAPAGEORGIOU, Constantine P.; OREN, Michael; POGGIO, Tomaso. A general framework for object detection. *In: Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*. IEEE, 1998. p. 555-562.

PEREIRA, Rafael Cardoso. **Técnica de rastreamento e perseguição de alvo utilizando o algoritmo Haar Cascade aplicada a robôs terrestres com restrições de movimento**. 2017. 105 f. Dissertação (Mestrado em Engenharia Mecatrônica) – Universidade Federal do Rio Grande do Norte, Natal, Brasil, 2017.

QUEIROZ, José Eustáquio Rangel de; GOMES, Herman Martins. Introdução ao processamento digital de imagens. **Rita**, v. 13, n. 2, p. 11-42, 2006.

RAMOS, André. Fisiologia da visão, um estudo sobre o "ver" e o "enxergar". **Pontifícia Universidade Católica do Rio de Janeiro - PUC**, Rio de Janeiro, 2006.

REZAEI, Mahdi. **Creating a cascade of haar like classifiers Step by step**. 2014. Disponível em: https://www.cs.auckland.ac.nz/~m.rezaei/Tutorials/Creating_a_Cascade_of_Haar-Like_Classifiers_Step_by_Step.pdf. Acesso em: 03 jul. 2020.

ROAT, Melvin I. **Pinguecula and Pterygium**. 2019. Disponível em: <https://www.msdmannuals.com/home/eye-disorders/conjunctival-and-scleral-disorders/pinguecula-and-ptyerygium>. Acesso em: 26 set. 2020.

RODRIGUES, Andreia Sias *et al.* Um Estudo de Mapeamento Sistemático sobre Metodologias de Avaliação em Interação Humano-Computador voltadas à Tecnologia Assistiva com foco em Pessoas com Deficiência Motora. **iSys-Brazilian Journal of Information Systems**, v. 11, n. 3, p. 90-126, 2018.

RYUMIN, Dmitry *et al.* Vision-Based Assistive Systems for Deaf and Hearing Impaired People. *In: Computer Vision in Advanced Control Systems-5*. Springer, Cham, 2020. p. 197-223.

SANTOS, Rafael Lopes; ABRANTES, Arnaldo; JORGE, Pedro Mendes. EYE GAZE TRACKING SYSTEM FOR ADAPTED HUMAN-COMPUTER INTERFACE. **i-ETC: ISEL Academic Journal of Electronics Telecommunications and Computers**, v. 3, n. 1, p. 1, 2017.

SANTOS, Renata Ferreira dos *et al.* Tecnologia assistiva e suas relações com a qualidade de vida de pessoas com deficiência. **Revista de Terapia Ocupacional da Universidade de São Paulo**, v. 28, n. 1, p. 54-62, 2017.

SANTOS, Vinicius Bitencourt dos *et al.* MOUSE OCULAR PARA PESSOAS COM MOVIMENTOS LIMITADOS CAUSADOS POR PROBLEMAS CERVICAIS E/OU CEREBRAIS. *In: Colloquium Exactarum*, v. 7, n. 2, p. 89-101, 2015.

SCURI, Antonio Escaño. Fundamentos da imagem digital. **Pontifícia Universidade Católica do Rio de Janeiro**, p. 13, 1999.

SHAFIQUE, Muhammad. Awais; HATO, Eiji. Use of acceleration data for transportation mode prediction. **Transportation**, v. 42, n. 1, p. 163-188, 2014.

SILVA, Julio Batista. **Explorando o algoritmo de Viola-Jones na detecção e reconhecimento facial**. 2018. 91 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia da Computação) – Universidade Federal de São Carlos, São Carlos, 2018.

SOUZA, Matheus Luiz Anderle de. **Um estudo de técnicas de detecção de olhos utilizando Visão Computacional**. 2019. 56 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Mecatrônica) – Universidade Federal de Santa Catarina, Joinville, 2019.

TRIPATHI, Abhishek Kumar; MUKHOPADHYAY, Sudipta; DHARA, Ashis Kumar. Performance metrics for image contrast. *In: 2011 International Conference on Image Information Processing*. IEEE, 2011. p. 1-4.

UDDIN, Mohammad Salah; AKHI, Afroza Yesmin. Horse detection using haar like features. *In: International Journal of Computer Theory and Engineering*, v. 8, n. 5, p. 415-418, 2016.

VIOLA, Paul; JONES, Michael J. Rapid object detection using a boosted cascade of simple features. *In: Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition - CVPR 2001*. IEEE, 2001.

VIOLA, Paul; JONES, Michael J. Robust real-time face detection. *In: International journal of computer vision*, v. 57, n. 2, p. 137-154, 2004.

VOLCHKO, Jeannine. **Visible Light Spectrum: From a Lighting Manufacturer's Perspective**. 2018. Disponível em: <https://www.lumitex.com/blog/visible-light-spectrum>. Acesso em: 24 jun. 2020.

WATANABE, Daniel Takeshi. **Sistema de visão computacional para a contagem de sementes de soja**. 2019. 49 f. Monografia (Especialização em Automação Industrial) – Universidade Tecnológica Federal do Paraná, Curitiba, 2019.

WESTLAND, Stephen; CHEUNG, Vien. RGB Systems. *In: Chen J., Cranton W., Fihn M. (eds) Handbook of Visual Display Technology*. Springer, Berlin, Heidelberg, 2015.

YONSEI UNIVERSITY, Computational Intelligence and Photography Lab. **Real and Fake Face detection**. 2019. Disponível em: <https://www.kaggle.com/ciplab/real-and-fake-face-detection>. Acesso em: 04 de jul. 2019.

ZAGHETTO, Alexandre *et al.* Facial Biometrics Using SIFT/SURF in Comparison to Eigenfaces. *In: XIII Workshop de Visão Computacional*. 2017.

ZHANG, Xucong *et al.* Appearance-based gaze estimation in the wild. *In: Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015. p. 4511-4520.