

CURSO DE CIÊNCIA DA COMPUTAÇÃO – TCC	
(X) PRÉ-PROJETO () PROJETO	ANO/SEMESTRE: 2022/2

APLICAÇÃO DE RASTREAMENTO OCULAR POR INTERMÉDIO DE WEBCAM UTILIZANDO VISÃO COMPUTACIONAL

Luis Augusto Kühn

Prof. Aurélio Faustino Hoppe – Orientador

1 INTRODUÇÃO

Segundo Duchowski (2017), o rastreamento ocular (em inglês, *eye tracking*) consiste em uma tecnologia para monitorar a posição e os movimentos dos olhos durante a realização de tarefas envolvendo a observação de estímulos visuais. Ainda de acordo com o autor, o rastreamento ocular desempenha um papel significativo na compreensão do comportamento humano e na Interação Humano-Computador (IHC).

Ansari *et al.* (2021), ressalta que atualmente existem muitos métodos que realizam o rastreamento ocular, sendo que a maioria das abordagens necessitam de um hardware adicional para aquisição de dados. Porém, em sua maioria apresentam um custo elevado. Por outro lado, há um número crescente quanto ao desenvolvimento de aplicações utilizando do rastreamento ocular. Ainda segundo os autores, as principais técnicas de estimativa ocular utilizam um hardware com luz infravermelha refletida na íris para rastrear o ponto de observação. Para isso, a câmera precisava de condições específicas para calcular os pontos do olhar, como estabilizar a cabeça de uma pessoa e controlar as condições de luz dentre outras limitações.

De acordo com Al-Btoush *et al.* (2019), aplicações de rastreamento ocular incluem a capacidade de controlar a posição do cursor em relação a posição dos olhos, permitindo identificar pontos focais do olhar ou utilizando como dispositivos de entrada como mouse e teclado, assim proporcionando maior acessibilidade aos dispositivos. Ainda segundo os autores, o rastreamento ocular apresenta grande importância pois podem ser utilizadas para diversas aplicações na área da saúde, engenharia de software, marketing etc. Na psicologia, por exemplo, a aplicação de métodos de rastreamento ocular permite investigar, em tempo real, os processos cognitivos (percepção, atenção, memória, tomada de decisão, resolução de problemas, etc.) envolvidos durante a realização de testes e tarefas. A partir disso é possível uma melhor compreensão sobre os processos mentais que levaram à determinada resposta, além de investigar quais áreas do teste (estímulos, alvos, opções de respostas, etc.) chamaram mais atenção (VAN HOOFT; BORN, 2012).

Duchowski (2017), também destaca que o rastreamento por vídeo, usado pelos rastreadores mais comumente encontrados, é realizado, em geral, com base na detecção de reflexos da pupila e da córnea gerados a partir da emissão de luz infravermelha. A imagem dos olhos é então capturada em intervalos razoavelmente regulares por uma câmera. Essas amostras de imagem são analisadas e, a partir de cálculos geométricos, é inferido o direcionamento do olhar. Al-Btoush *et al.* (2019) apontam que estudos recentes sobre rastreamento ocular demonstram que a utilização de técnicas de visão computacional e aprendizado de máquina podem estabelecer de forma mais precisa a posição ocular. Assim como Ansari *et al.* (2021) utilizaram uma Convolutional Neural Network (CNN) modificada para realizar um estudo que demonstra uma viabilidade. Além disso, também há uma necessidade de aplicações que reduzam o custo e estejam disponíveis de forma acessível.

Diante deste cenário, este trabalho propõe o desenvolvimento de uma aplicação de rastreamento ocular que seja acessível através de uma aplicação com acesso a uma webcam.

1.1 OBJETIVOS

O objetivo deste trabalho é desenvolver uma aplicação que seja capaz de rastrear a posição ocular através de uma webcam.

Os objetivos específicos são:

- a) analisar e comparar a precisão dos algoritmos utilizados na detecção facial e ocular assim como, em relação ao rastreamento/mapeamento das movimentações dos olhos;
- b) identificar em quais condições/ambientes é possível realizar o rastreamento ocular;
- c) disponibilizar uma aplicação na plataforma web para facilidade de acesso, cadastro de testes e visualização dos resultados;
- d) avaliar a possibilidade de adoção da aplicação por profissionais na saúde.

2 TRABALHOS CORRELATOS

Nesta seção são apresentados trabalhos que possuem semelhanças ao trabalho proposto. A seção 2.1 apresenta o desenvolvimento de um protótipo para auxiliar pessoas com mobilidade reduzida a interagir com o meio externo através de uma câmera de uso geral (ADRIANO *et al.*, 2020). Na seção 2.2, aborda-se o

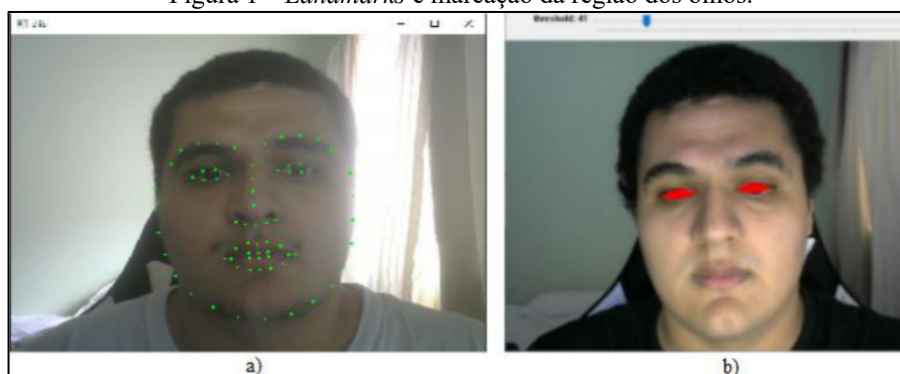
desenvolvimento de um método de estimativa da posição do olho, utilizando *Convolutional Neural Network* (CNN) (ANSARI, *et al.* (2021)). Por fim, a seção 2.3 discorre sobre a utilização de um método para localização dos olhos a partir da posição (x, y) (AL-BTOUSH *et al.*, 2019).

2.1 SISTEMA DE RASTREAMENTO OCULAR NÃO INTRUSIVO UTILIZANDO A BIBLIOTECA OPENCV

Adriano *et al.* (2020) tinham como objetivo desenvolver um protótipo para rastreamento ocular utilizando a biblioteca GazeML e uma *webcam* de baixo custo. Para isso, os autores utilizaram a linguagem Python, em conjunto com a biblioteca OpenCV para captura e manipulação de *frames* obtidos através de uma webcam Logitech c270 que captura sessenta frames por segundo, posicionada em frente ao usuário na parte superior do monitor. Além disso, segundo os autores, também foram utilizados os algoritmos de detecção facial HaarCascade, a biblioteca Dlib e o Pygame para animação.

Segundo Adriano *et al.* (2020), o algoritmo HaarCascade apresentou que deve dificuldades para identificar a posição dos olhos utilizando esse algoritmo, por isso, utilizou-se a biblioteca Dlib para detecção dos 68 pontos da face juntamente com a implementação do projeto de código aberto GazeML para identificação das *landmarks* que permitem identificar características da face baseada em pontos. A Figura 1 exemplifica a marcação da região dos olhos via *landMarks*.

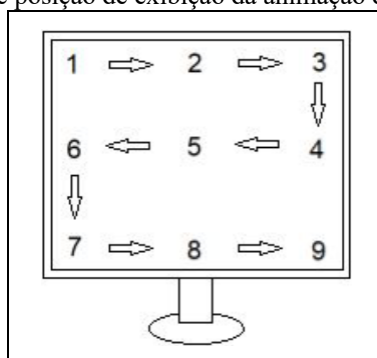
Figura 1 – *Landmarks* e marcação da região dos olhos.



Fonte: Adriano *et al.* (2020).

Para realizar a calibração, Adriano *et al.* (2020) estabeleceram um conjunto de pontos dispostos na tela do computador do usuário Figura 2, ao qual deveria fixar o olho durante 2 segundos. Para cada ponto representado, realizava-se um cálculo para identificar a diferença entre a íris e o centro do olho.

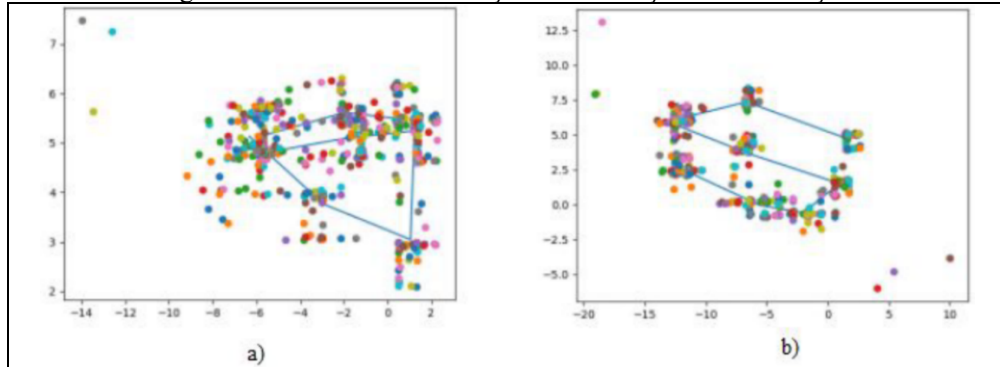
Figura 2 – Ordem e posição de exibição da animação durante a calibração.



Fonte: Adriano *et al.* (2020).

A matriz de calibração foi construída através da captura de sessenta *frames* em cada uma das 9 posições apontadas na Figura 2 é calculada através do produto entre a matriz inversa gerada anteriormente e as posições no eixo x utilizadas para exibição da animação, determinando o coeficiente x, o mesmo aplica-se para o coeficiente do eixo y, o resultado é a matriz de calibração que ao calcular o produto com uma entrada, retornará a estimativa de uma coordenada no monitor do usuário e dos sessenta frames é calculada a média aritmética, resultando em uma matriz de nove por dois. A Figura 3 apresenta os resultados da calibração com alterações na iluminação.

Figura 3 – Resultado da calibração com alterações na iluminação.



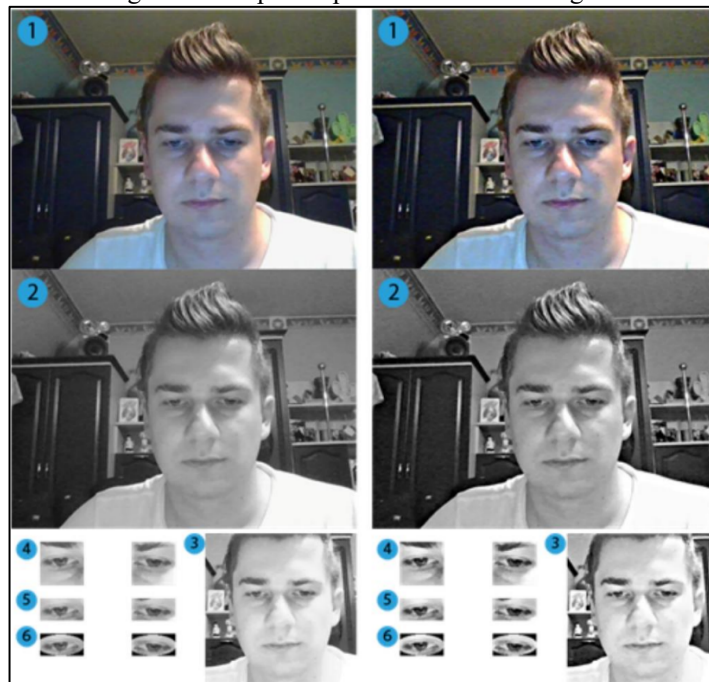
Fonte: Adriano *et al.* (2020).

Adriano *et al.* (2020) realizaram testes considerando duas situações: (i) com uma lâmpada no teto e em um ambiente fechado e (ii) utilizando uma iluminação auxiliar em frente ao usuário. Ao total, foram registrados 540 pontos com cores diferentes para facilitar a visualização daqueles que possuíam coordenadas próximas (similares). Além disso, os autores também adicionaram linhas em azul para estabelecer os pontos médios a cada sessenta frames. Na Figura 3 item (a), pode-se perceber que utilizando a iluminação de teto em um ambiente fechado se obteve baixa precisão apresentando pontos inconsistentes no gráfico, já na Figura 3 item (b) utilizando iluminação auxiliar obteve uma melhor precisão apresentando um gráfico semelhante a uma animação de forma espelhada, pode-se concluir que cada calibração apresentou um resultado único tendo como, como grande influência a iluminação, mas também a distância e posição do rosto. Por fim, Adriano *et al.* (2020) relatam que os algoritmos para rastreamento ocular dependem de uma calibração eficiente para alcançar uma boa precisão.

2.2 GAZE TRACKING USING AN UNMODIFIED WEB CAMERA AND CONVOLUTIONAL NEURAL NETWORK

Ansari *et al.* (2021) tinham como objetivo analisar e testar um conjunto de dados coletados por pessoas utilizando a câmera do seu próprio *laptop*, assim permitindo a aquisição de imagens em condições reais. O único ponto de atenção solicitado aos usuários era em relação à iluminação do ambiente. Para a coleta dos dados, os autores utilizaram o EyeTrackerDataCollector, que é um aplicativo para desktop que registra imagens de uma câmera. No pré-processamento e processamento da imagem, os autores utilizaram a biblioteca OpenCV. Inicialmente, eles converteram a imagem em escala de cinza, detectaram a face através do classificador Viola-Jones e os olhos utilizando o HaarCascade como demonstrado na Figura 4.

Figura 4 – Etapas do processamento da imagem.

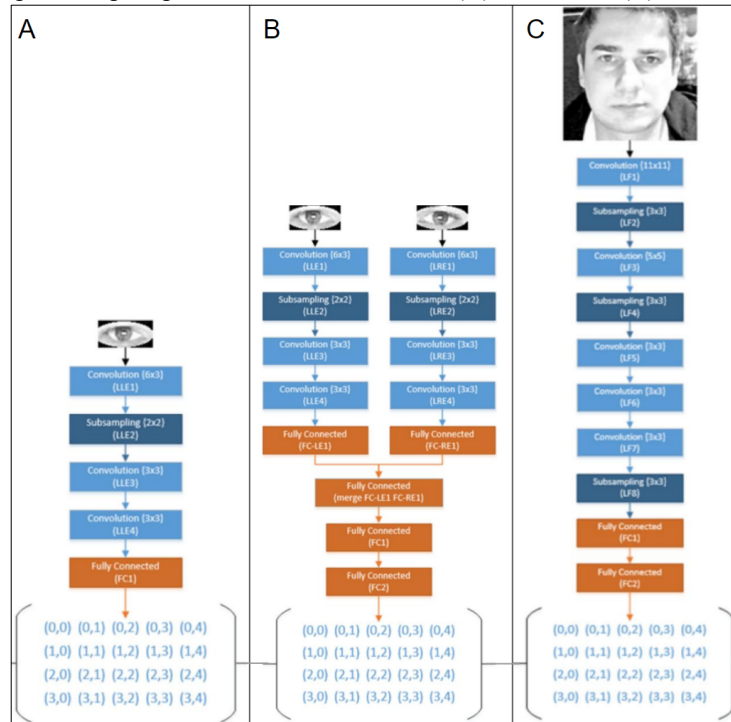


Fonte: Ansari *et al.* (2021).

Para o treinamento da rede neural, Ansari *et al.* (2021) optaram por avaliar a performance 3 arquiteturas diferentes, buscando entender qual a melhor forma para processar as imagens. A primeira arquitetura foi treinada

considerando apenas um dos olhos. A Figura 5 item (A) demonstra que a arquitetura contém três camadas convolucionais, uma de subamostragem com um tamanho de filtro 2×2 e uma camada totalmente conectada. Na primeira camada convolucional (LLE1) possui 96 filtros, a camada LLE3 possui 384 filtros e LLE4 possui 256 filtros, a camada totalmente conectada (FC1) possui 256 neurônios e na última camada gerada retorna um vetor de 20 valores, visando detectar apenas um dos olhos.

Figura 5 – Arquitetura para processamento de um olho (A), dois olhos (B) e Face completa (C).



Fonte: Ansari *et al.* (2021).

Na segunda arquitetura, construiu-se um modelo que recebe como entrada os dados os olhos direito e esquerdo, conforme exhibe a Figura 5 item (B). Para isso, a rede consiste em três camadas convolucionais, uma camada de subamostragem 2×2 , uma camada de concatenação e três camadas totalmente conectadas. As camadas totalmente conectadas consistem em uma camada para cada imagem de entrada, ou seja, FC-LE1 e FC-RE1 com 1024 neurônios, a camada concatenada e duas camadas: FC1 com 2048 neurônios e FC2 com 1024 neurônios. A saída da rede consiste em um conjunto de vinte pontos que representam a posição da área ao qual está sendo focalizado.

Ansari *et al.* (2021) também treinaram uma terceira rede que é detalhada na Figura 5 item (C), utilizando como entrada a imagem de toda a face. Ela é composta por cinco camadas convolucionais, três camadas de subamostragem com tamanho de filtro 3×3 e duas camadas totalmente conectadas FC1 e FC2. O número de neurônios das camadas totalmente conectadas foi de 4096 para FC1 e 1000 para FC2.

Para o treinamento da rede, Ansari *et al.* (2021) utilizaram dois conjuntos de dados, o primeiro consiste em 6.000 imagens e foi adquirida através do *click* em 20 pontos que foram exibidos na tela, de forma uniforme, em uma grade cinco por quatro. A segunda base também se utilizou o sistema de pontos, mas utilizando a amostra de quatro outras pessoas. No qual um deles realizou o teste duas vezes, uma vez utilizando óculos e a outra com lentes de contato.

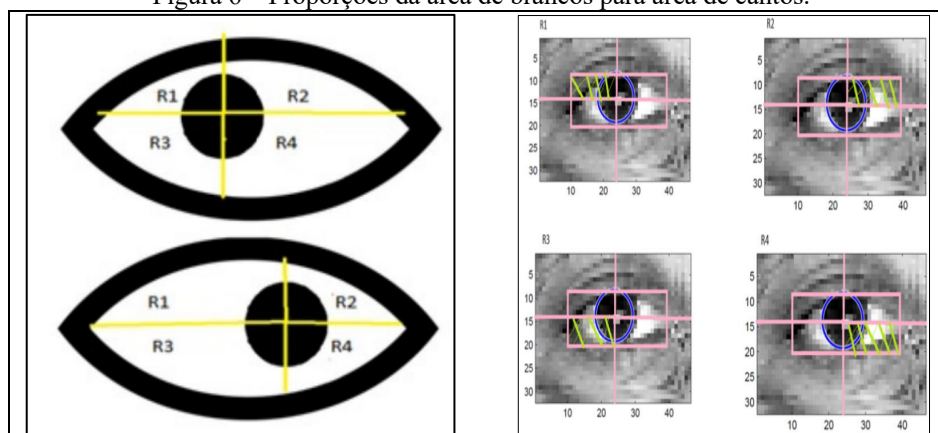
Durante os testes realizados, Ansari *et al.* (2021) observaram que ao melhorar a nitidez, aumentou-se a taxa de classificação de 77% para 81%. A partir disso, os autores apontam que a arquitetura treinada apenas considerando o olho esquerdo teve resultados acima de 88% de precisão. Já a arquitetura para detectar os dois olhos, não obteve resultados satisfatórios e que as 70 épocas utilizadas no treino não foram suficientes. Ansari *et al.* (2021) relatam que a arquitetura que utilizava a face completa apresentou precisão de 81,81%. Resultado considerado razoável, entretanto, os autores ressaltam o alto custo computacional para realizar o treinamento. Além disso, eles afirmam que a quantidade de dados irrelevantes na imagem não interferiram no resultado final.

Adriano *et al.* (2020) concluem que através de uma rede CNN é possível obter resultados que podem ser utilizados em aplicações práticas em ambiente real. Eles também apontam que o ponto de olhar não foi calculado através de uma regressão. Além disso, o número de participantes e a quantidade de dados eram limitados.

2.3 NEW FEATURES FOR EYE-TRACKING SYSTEMS: PRELIMINARY RESULTS

Al-btoush *et al.* (2019 p. 179) tinham como objetivo localizar a posição da pupila e definir o ponto focal. Para isso, os autores utilizaram duas estratégias: a primeira utiliza a posição (x, y) da pupila para definir o ponto focal e, o segundo, calcula a posição da pupila utilizando as áreas da esclera, buscando encontrar a parte branca em cada um dos cantos do olho R1, R2, R3 e R4, conforme demonstra a Figura 6.

Figura 6 – Proporções da área de brancos para área de cantos.



Fonte: Al-btoush *et al.* (2019).

Al-btoush *et al.* (2019 p.181) utilizaram como detector de face, o método de detecção Viola-Jones. Para a detecção da pupila, utilizou-se a transformada circular de Hough. Além disso, também se utilizou a plataforma WEKA de mineração de dados para realizar os experimentos utilizando uma série de algoritmos de aprendizado de máquina já disponíveis na plataforma. A ferramenta MATLAB foi utilizada para calcular a quantidade de pixels brancos em cada uma das partes R1, R2, R3, R4 da esclera. Os autores testaram os algoritmos de aprendizado de máquina *Multilayer perceptron* (MLP), classificador de Árvore de decisão J48 e *Random Forest* (RF).

De acordo com Al-btoush *et al.* (2019 p.181) foram realizados 3 experimentos utilizando as 500 imagens que foram adquiridas na observação de 10 usuários (7 homens e 3 mulheres). O primeiro experimento utilizou a localização (x, y) da pupila. O segundo utilizou as características dos pontos R1, R2, R3 e R4 que são os referentes a divisão da esclera como mostrado na Figura 6. E, o terceiro, juntou ambas as características, ao qual não apresentou bons resultados em formato multiusuário, devido a diversas características: (i) comprimentos dos cílios, (ii) aparecimento/desaparecimento do arco do olho, (iii) formato dos olhos, (iv) sombras, entre outras características. O Quadro 1 apresenta a precisão de cada algoritmo a partir das características observadas.

Quadro 1 – Resultados da precisão dos algoritmos de classificação

Testes	MLP	J48	RF
X, Y	95.6%	96.2%	97%
R1, R2, R3, R4	55.4%	60.6%	65.6%
X, Y, R1, R2, R3, R4	95.2%	95.6%	97.2%

Fonte: Al-btoush *et al.* (2019, p. 182).

A partir Quadro 1, pode-se observar que os resultados utilizando ambos os mecanismos de detecção aumentaram a precisão do sistema. De acordo com os autores, a principal motivo que influenciou os resultados foi a dificuldade para detectar a esclera. Por fim, Al-btoush *et al.* (2019 p. 181) concluem que os resultados experimentais usando alguns algoritmos de aprendizado de máquina demonstraram que os novos recursos são dependentes do usuário e, portanto, em seu formato atual não podem ser usados para um sistema de rastreamento ocular multiusuário. Além disso, também se observou que os novos recursos podem ser utilizados para aprimorar os sistemas tradicionais e, que no processo de pré-processamento, poderia ser realizado o realce da imagem, remoção de vermelhidão da esclera e cílios.

3 PROPOSTA

A seguir é apresentada a justificativa para o desenvolvimento desse trabalho, os principais requisitos e a metodologia de desenvolvimento que será utilizada. Também são relacionados os assuntos e as fontes bibliográficas que irão fundamentar o estudo proposto.

3.1 JUSTIFICATIVA

No Quadro 2 é apresentado um comparativo entre os trabalhos correlatos. As linhas representam as características e as colunas os trabalhos.

Quadro 2 – Comparativo dos trabalhos correlatos

Trabalhos Correlatos Características	Adriano <i>et al.</i> (2020)	Ansari <i>et al.</i> (2021)	Al-Btoush <i>et al.</i> (2019)
Característica do método	Não Invasivo	Não Invasivo	Não Invasivo
Forma de detecção da face	Cascata de Haar/Dlib	Viola-Jones	Viola-Jones
Deteção da pupila	Cascata de Haar	Cascata de Haar	Transformada circular de Hough
Forma de análise	Olho esquerdo e direito	Olho esquerdo e direito, face completa	Olho esquerdo
Técnica de análise utilizada	GazeMI	CNN	MLP, J48 e RF

Fonte: elaborado pelo autor.

Conforme pode ser observado no Quadro 2, a detecção de face é um fator bastante relevante que pode ser observado em todos os trabalhos. Adriano *et al.* (2020) utilizaram o algoritmo HaarCascade/Viola-Jones, assim como a utilização da biblioteca Dlib para criação dos *landmarks* e aprimorar a precisão da detecção. Na análise, os autores optaram pela utilização dos olhos direito e esquerdo para detecção da posição (x, y) da pupila. Os autores também realizaram uma análise detalhada para identificar a melhor forma para realizar a detecção, no qual foram testadas três formas de detecção do olho direito e esquerdo, apenas do esquerdo e da face completa. Adriano *et al.* (2020) concluíram que a melhor forma seria considerando apenas o olho esquerdo. Além disso, também apontam que devido ao baixo poder de processamento, não conseguiram estabelecer a eficiência utilizando a detecção da face completa.

Na pesquisa realizada por Ansari *et al.* (2021) realizou-se testes em arquiteturas com diferentes enquadramentos, buscando realizar o rastreamento ocular e ponto focal de forma precisa. Os autores utilizaram dados capturados por pessoas utilizando a câmera do seu próprio *laptop*.

Al-Btoush *et al.* (2019) verificaram a semelhança a partir do olho esquerdo. Os autores, apresentaram uma nova metodologia para calcular a localização da pupila, considerando a detecção dos pontos R1, R2, R3, R4, que são os quatro cantos da esclera. Al-Btoush *et al.* (2019) também utilizaram como forma de detecção o (x, y) da pupila assim como Adriano *et al.* (2020) e Ansari *et al.* (2021). Os autores relatam a obtenção de 97.2% através do algoritmo de *Random Forest*.

Diante deste contexto, este trabalho mostra-se relevante, pois pretende desenvolver uma aplicação que permita identificação mais precisa de rastreamento ocular através de entradas em tempo real utilizando uma webcam. Para isso, serão utilizadas técnicas de visão computacional para o tratamento da imagem e algoritmos de rede neural que permitam uma melhor precisão ao identificar o local onde o usuário está olhando. Além disso, o protótipo será disponibilizado através de uma aplicação web, buscando facilitar o acesso por profissionais da saúde que pretendam utilizar a ferramenta para tratamento de Déficit de atenção, Dislexia e Autismo.

3.2 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

Esta seção mostrará os Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF) para a aplicação de rastreamento ocular. Os requisitos do trabalho estão divididos em duas partes: requisitos da aplicação web e requisitos do módulo de reconhecimento.

A aplicação web deverá:

- manter o cadastro de usuários (RF);
- permitir o login de usuários já cadastrados (RF);
- permitir ao usuário a criação de um teste de rastreamento ocular (RF);
- permitir ao usuário informar uma lista de imagens/vídeos (RF);
- permitir ao usuário possa vincular e visualizar imagens/vídeos a um teste (RF);
- realizar o *eye-tracking* do usuário enquanto assiste uma apresentação de uma sequência de exibição de imagens/vídeos em ordem e posição aleatórias (RF);
- exibir um mapa visual das coordenadas obtidas pelo *eye-tracking* a partir das imagens/vídeos apresentados (RF);
- exportar este mapa de preferência visual para PDF (RF);
- utilizar o ambiente de desenvolvimento Visual Studio Code (RNF);
- desenvolver a aplicação web utilizando NodeJS no *backend* e React no *frontend* (RNF);
- utilizar o banco de dados MySQL para persistir os dados (RNF).

O módulo de rastreamento ocular a ser desenvolvido deverá:

- realizar uma calibração para estabelecer o perfil de movimentação dos olhos (RF);
- realizar a detecção facial e a identificação da região dos olhos utilizando aprendizado de máquina (RF);
- extrair a localização da pupila e seu posicionamento utilizando visão computacional (RF);
- realizar rastreamento ocular através de uma webcam (RNF);
- ser desenvolvida na linguagem de programação Python (RNF);
- utilizar as bibliotecas OpenCV, Keras e scikit-learn (RNF).

3.3 METODOLOGIA

O trabalho será desenvolvido observando as seguintes etapas:

- levantamento bibliográfico: pesquisar sobre rastreamento e mapeamento da posição ocular, redes neurais e trabalhos correlatos;
- elicitação de requisitos: baseando-se no levantamento bibliográfico, refinar os requisitos propostos para a aplicação proposta;
- especificação da aplicação web: utilizar a ferramenta Enterprise Architect (EA) para elaborar os diagramas de caso de uso e de atividades de acordo com a Unified Modeling Language (UML);
- implementação: a partir do item (c) implementar a aplicação Web utilizando a NodeJS no *backend* e React no *frontend*;
- testes da aplicação web: elaborar testes para validar a usabilidade da aplicação assim como, apresentar e realizar testes com os profissionais da saúde para averiguar sua aceitação e utilidade.
- coleta e montagem da base de imagens: coletar as imagens que serão usadas nos testes do método proposto. Serão pesquisadas opções de bases de dados públicas e também será avaliada a coleta de imagens em um ambiente específico controlado;
- pesquisa e escolha do algoritmo de detecção facial e ocular: pesquisar os principais algoritmos de aprendizado de máquina para a detecção facial e ocular, escolhendo o mais adequado para o desenvolvimento do trabalho;
- desenvolvimento do modelo de detecção facial e ocular: a partir do item (g), realizar a implementação da arquitetura da rede neural utilizando Python e a biblioteca Keras e scikit-learn;
- pesquisa e escolha do algoritmo de rastreamento dos olhos: pesquisar os principais algoritmos de rastreamento de olhos, escolhendo o mais adequado para o desenvolvimento do trabalho;
- levantamento de formas para calcular a movimentação da pupila: pesquisar métodos para mapear a posição pupila de acordo com o que o usuário está visualizando em tela;
- desenvolvimento do método: desenvolver o método de detecção e mapeamento da movimentação dos olhos utilizando a linguagem de programação Python e a biblioteca OpenCV;
- testes da detecção e mapeamento: em paralelo ao desenvolvimento, verificar a assertividade do método proposto a partir do percentual do rastreamento e mapeamento correto, e caso necessário, alterar os requisitos para atender o problema a ser resolvido. Além disso, também serão realizados testes em diferentes ambientes, considerando diferentes distâncias e iluminação, para estabelecer em qual deles o método possui maior precisão.

As etapas serão realizadas nos períodos relacionados no Quadro 3.

Quadro 3 – Cronograma de atividades a serem realizadas

etapas / quinzenas	2022				2023											
	nov.		dez.		jan.		fev.		mar.		abr.		maio		jun.	
	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
levantamento bibliográfico																
elicitação de requisitos																
especificação da aplicação web																
implementação																
testes da aplicação web																
coleta e montagem da base de imagens																
pesquisa e escolha do algoritmo de detecção facial e ocular																
desenvolvimento do modelo de detecção facial e ocular																
pesquisa e escolha do algoritmo de rastreamento dos olhos																
levantamento de formas para calcular a movimentação da pupila																
desenvolvimento do método																
testes da detecção e mapeamento																

Fonte: elaborado pelo autor.

4 REVISÃO BIBLIOGRÁFICA

Esta seção descreve brevemente sobre os assuntos que fundamentarão o estudo a ser realizado: rastreamento e mapeamento da posição ocular e redes neurais.

Watanabe (2013) afirma que o rastreamento ocular é o processo de identificar a movimentação do globo ocular de um indivíduo e para essa detecção são apresentados três passos: (i) localização da face do usuário, (ii) detectar a região dos olhos (linha visual) e extrair as características do olho, (iii) rastrear os componentes do olho numa sequência de imagens. A saída de um sistema de rastreamento ocular é o cálculo realizado pelo *Point Of Regard* (POR), ao qual é determinado pelo mapeamento das coordenadas (x, y) do olhar do usuário.

De acordo com Barreto (2012), as Fixações do olhar são momentos em que os olhos estão fixos assimilando ou identificando alguma informação. Esse padrão pode ter diferentes significados dependendo do contexto, na navegação em uma página web a maior frequência de fixação em uma área específica pode ser classificado como um maior interesse na área a ser observada ou também pode estar apresentando um ponto onde o usuário demonstra dificuldade para interpretar a informação, um número grande de fixações pode apresentar uma dificuldade na busca ou reconhecimento de um item no destino.

Barreto (2012) apresentou algumas métricas para o reconhecimento do padrão de olhar.

- a) Não codificação: a não codificação ocorre durante as sacadas que é movimentar o olho de uma posição visual para outra, ela não nos permite saber muitas informações sobre a complexidade ou relevância de um objeto.
- b) Regressão (sacada regressiva): é uma sacada que volta a posição anterior, como na leitura de um texto aonde os olhos voltam para a linha anterior, pode ser utilizado como medição de dificuldade durante a decodificação da informação.
- c) Duração do olhar (*gaze duration, dwell, fixation cluster ou fixation cycle*): é caracterizada por diversas fixações consecutivas de curta duração em uma área de interesse específica, e a fixação que ocorre fora da área de interesse demonstra o fim do olhar.
- d) “Scanpaths”(sequência de fixações): é caracterizada pela transição de uma área de interesse para outra e pode ser utilizada para identificar a eficiência da disposição dos elementos.
- e) Taxa de intermitência (piscar de olhos) e tamanho da pupila: a taxa de intermitência e o tamanho da pupila podem ser usados como um índice de carga cognitiva. Uma taxa de intermitências menor apresenta uma maior carga de trabalho já uma menor pode ser caracterizada por uma fadiga.
- f) Número total de fixações: número de fixações pode apresentar uma menor eficiência na disposição dos elementos da interface.
- g) Número de fixações sobre uma área de interesse: um maior número de fixações pode indicar maior importância da informação para o usuário.

Segundo Barreto (2012), utilizando esses padrões de olhar pode ser possível identificar a uma ação que permite aprimorar uma interface ou preferência visual do indivíduo.

REFERÊNCIAS

- ADRIANO, Bruno *et al.* Sistema de rastreamento ocular não intrusivo utilizando a biblioteca opencv. In: **11º congresso de iniciação científica e tecnológica do ifsp**. 2020. Acesso em: 2 ago. 2022.
- AL-BTOUSH, Audi I. *et al.* New features for eye-tracking systems: Preliminary results. In: **2019 10th international conference on information and communication systems (icics)**. *ieee*, 2019. p. 179-184. Acesso em: 10 ago. 2022.
- ANSARI, Mohd Faizan *et al.* **Gaze Tracking Using an Unmodified Web Camera and Convolutional Neural Network**. *Applied Sciences*, v. 11, n. 19, p. 9068, 2021. Acesso em: 10 ago. 2022.
- BARRETO, Ana Margarida. Eye tracking como método de investigação aplicado às ciências da comunicação. **Revista Comunicando**, [s. l.], v1, n1, dezembro 2012. Disponível em: <https://revistas.sopcom.pt/index.php/comunicando/article/view/126>. Acesso em: 10 out. 2022
- DUCHOWSKI, Andrew. T. **Eye Tracking Methodology: Theory and Practice** (3rd ed.). Springer-Verlag. London 2017. doi: 10.1007/978-3-319-57883-5
- VAN HOOFT, Edwin. A.; BORN, Marise. P. **Intentional response distortion on personality tests: using eye-tracking to understand response processes when faking**. *The Journal of applied psychology*, 2012. Mar 97(2), 301–316. <https://doi.org/10.1037/a002571>
- WATANABE, Marcus Vinícius Hideki. **Eye tracking e suas aplicações**. 2013. 56f. Monografia (Bacharel em Ciência da Computação) - Departamento de Computação, Universidade Estadual de Londrina, Londrina, 2013. Acesso em: 12 out. 2022

FORMULÁRIO DE AVALIAÇÃO BCC – PROFESSOR AVALIADOR – PRÉ-PROJETO

Avaliador(a): Marcel Hugo

Atenção: quando o avaliador marcar algum item como atende parcialmente ou não atende, deve obrigatoriamente indicar os motivos no texto, para que o aluno saiba o porquê da avaliação.

ASPECTOS AVALIADOS		Atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO O tema de pesquisa está devidamente contextualizado/delimitado?			
	O problema está claramente formulado?			
	2. OBJETIVOS O objetivo principal está claramente definido e é passível de ser alcançado?			
	Os objetivos específicos são coerentes com o objetivo principal?			
	3. TRABALHOS CORRELATOS São apresentados trabalhos correlatos, bem como descritas as principais funcionalidades e os pontos fortes e fracos?			
	4. JUSTIFICATIVA Foi apresentado e discutido um quadro relacionando os trabalhos correlatos e suas principais funcionalidades com a proposta apresentada?			
	São apresentados argumentos científicos, técnicos ou metodológicos que justificam a proposta?			
	São apresentadas as contribuições teóricas, práticas ou sociais que justificam a proposta?			
	5. REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO Os requisitos funcionais e não funcionais foram claramente descritos?			
	6. METODOLOGIA Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?			
	Os métodos, recursos e o cronograma estão devidamente apresentados e são compatíveis com a metodologia proposta?			
	7. REVISÃO BIBLIOGRÁFICA Os assuntos apresentados são suficientes e têm relação com o tema do TCC?			
	As referências contemplam adequadamente os assuntos abordados (são indicadas obras atualizadas e as mais importantes da área)?			
ASPECTOS METODOLÓGICOS	8. LINGUAGEM USADA (redação) O texto completo é coerente e redigido corretamente em língua portuguesa, usando linguagem formal/científica?			
	A exposição do assunto é ordenada (as ideias estão bem encadeadas e a linguagem utilizada é clara)?			