



UEPB

**UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS VII – GOVERNADOR ANTÔNIO MARIZ
CENTRO DE CIÊNCIAS EXATAS E SOCIAIS APLICADAS
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

KAIQUE RIJKAARD DE SOUSA OLIVEIRA

**SISTEMA PARA DETECÇÃO DO PESTANEJAR UTILIZANDO VISÃO
COMPUTACIONAL PARA PREVENÇÃO DA SÍNDROME VISUAL DO
COMPUTADOR**

**PATOS
2019**

KAIQUE RIJKAARD DE SOUSA OLIVEIRA

**SISTEMA PARA DETECÇÃO DO PESTANEJAR UTILIZANDO VISÃO
COMPUTACIONAL PARA PREVENÇÃO DA SÍNDROME VISUAL DO
COMPUTADOR**

Trabalho de Conclusão de Curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de Bacharel em Ciência da Computação.

Área de concentração: Visão Computacional.

Orientadora: Prof. Dra. Jannayna Domingues Barros Filgueira

Co-orientador: Prof. Esp. Fábio Junior Francisco da Silva

**PATOS
2019**

É expressamente proibido a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano do trabalho.

O48s Oliveira, Kaique Rijkaard de Sousa.

Sistema para detecção do pestanejar utilizando visão computacional para prevenção da síndrome visual do computador [manuscrito] / Kaique Rijkaard de Sousa Oliveira. - 2019.

56 p. : il. colorido.

Digitado.

Trabalho de Conclusão de Curso (Graduação em Computação) - Universidade Estadual da Paraíba, Centro de Ciências Exatas e Sociais Aplicadas, 2019.

"Orientação : Profa. Dra. Jannayna Domingues Barros Filgueira, Coordenação do Curso de Computação - CCEA."

"Coorientação: Prof. Esp. Fábio Junior Francisco da Silva, Coordenação do Curso de Computação - CCEA."

1. Visão computacional. 2. Detecção facial. 3. Processamento de imagem. I. Título

21. ed. CDD 005.3

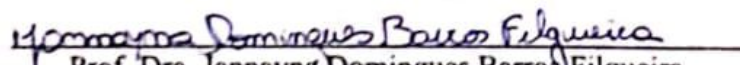
Kaique Rijkaard de S. Oliveira

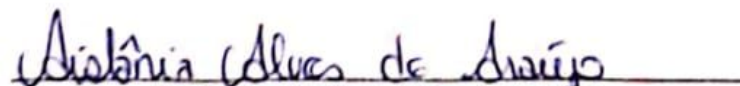
**SISTEMA PARA DETECÇÃO DO PESTANEJAR UTILIZANDO VISÃO
COMPUTACIONAL PARA PREVENÇÃO DA SÍNDROME VISUAL DO
COMPUTADOR**

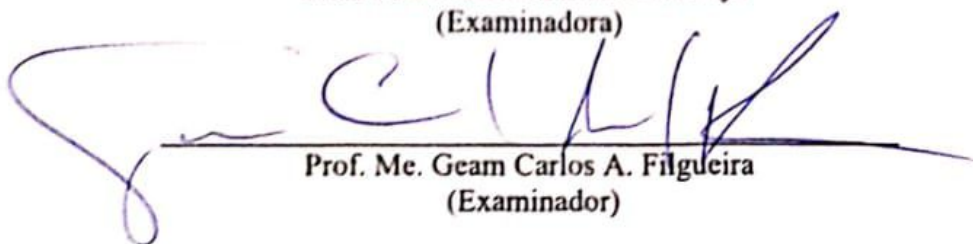
Trabalho de Conclusão de Curso apresentado ao
Curso de Bacharelado em Ciências da
Computação da Universidade Estadual da
Paraíba, em cumprimento à exigência para
obtenção do grau de Bacharel em Ciência da
Computação.

Aprovado em 25/11/2019

BANCA EXAMINADORA


Prof. Dra. Jannayna Domingues Barros Filgueira
(Orientador)


Prof. Me. Aislânia Alves de Araújo
(Examinadora)


Prof. Me. Geam Carlos A. Filgueira
(Examinador)

A minha mãe, família e amigos, pelo incentivo, suporte e amizade, DEDICO.

AGRADECIMENTOS

Agradeço primeiramente a Deus pela saúde e determinação que sempre ouvistes minhas súplicas, tornando possível continuar percorrendo meu caminho e vencendo os obstáculos do dia-a-dia para realizar meus sonhos, promovendo dessa forma mais essa realização em minha vida.

A minha mãe Iracema de Sousa Oliveira, as minhas tias Lindalva e Alzira e ao meu tio Agenor por todo amor e amizade, estando sempre presente na minha vida aconselhando e apoiando no decorrer deste caminho. Suas palavras e presenças me confortam.

Ao meu pai Geraldo da Silva Oliveira (*in memoriam*), embora fisicamente ausente, sentia sua presença ao meu lado, dando-me força para superar as dificuldades.

A minha família independente de sangue Maria José e Larissa que também sempre estiveram presentes na minha vida. E aos demais familiares que me ajudaram a superar as adversidades.

A minha namorada Vangerlândia, pelo apoio e carinho que me confortaram e ajudaram na superação dos obstáculos.

Aos amigos e colegas de classe, em especial Ricardo, Marinaldo, Luiz Henrique, Hoffmann, João Paulo e Felype que sempre me ajudaram e proporcionaram momentos de amizade durante este percurso.

A minha orientadora Jannayna, pela amizade e por direcionar minha vida acadêmica sempre incentivando e aconselhando. Ao meu co-orientador Fábio Júnior pela atenção, conselhos e amizade.

Aos professores da minha formação acadêmica por transmitir seus conhecimentos por meio das disciplinas e conselhos, bem como a todos os professores que contribuíram em meu percurso escolar do pré-escolar até o ensino médio.

Aos funcionários da UEPB pela presteza e atendimento quando nos foi necessário, como também aos demais funcionários que prestaram serviços e possibilitavam trilhar meu caminho.

A todos que de alguma forma contribuíram para a minha vida. Agradeço!

“Nós só podemos ver um pouco do futuro, mas o suficiente para perceber que há muito a fazer.”
(Alan Turing)

RESUMO

Os dispositivos digitais, tais como computadores, notebooks, smartphones e tablets, são frequentemente utilizados pelas pessoas para entretenimento, comunicação e trabalho. O uso excessivo destes dispositivos diminui a quantidade de piscadas do olho do usuário, diminuindo a lubrificação ou filtro lacrimal, causando a desidratação e contribuindo para a síndrome visual do computador. Os sintomas são fadiga visual, irritação ocular, secura ocular, dor nos olhos, dificuldade em focar, entre outros. Esta pesquisa aplica visão computacional com o objetivo de desenvolver uma aplicação com as técnicas de detecção facial para localizar a região dos olhos e a partir desta região identificar em tempo real o pestanejar do usuário. A aplicação monitora o pestanejar do usuário e cronometra o tempo que ele passa sem piscar, notificando-o no prazo estabelecido. Para validar a aplicação foram efetuados testes de modo a verificar a precisão do algoritmo, bem como os experimentos para analisar a intervenção da aplicação em alertar os voluntários de pestanejar. Os experimentos consistiram na leitura de uma notícia e em assistir um vídeo, ao mesmo tempo que a aplicação monitorava e lembrava o pestanejar dos voluntários. Os resultados obtidos foram um percentual de 92,13% de precisão da aplicação com base nos experimentos realizados.

Palavras-Chave: Visão Computacional. Detecção Facial. Processamento de Imagem.

ABSTRACT

Digital devices, such as computers, laptops, smartphones and tablets, are often used by people for entertainment, communication and work. Overuse of these devices decreases the amount of blinking of the user's eye, decreasing the lubrication or tear filter, causing dehydration and contributing to visual computer syndrome. Symptoms are visual fatigue, eye irritation, dry eye, eye pain, difficulty focusing, among others. This research applies computer vision in order to develop an application with facial detection techniques to locate the eye region and from this region identify in real time the user's blink. The application monitors the user's blink and times their unblinking time, notifying them on time. To validate the application, tests were performed to verify the accuracy of the algorithm, as well as experiments to analyze the intervention of the application to alert the volunteers to blink. The experiments consisted of reading a news story and watching a video while the application monitored and resembled the blinking of volunteers. The results obtained were a percentage of 92.13% of application accuracy based on the experiments performed.

Keywords: Computer vision. Facial Detection. Image Processing.

LISTA DE ILUSTRAÇÕES

Figura 1 –	Representação de uma imagem digital com sua subimagem e seus respectivos valores de intensidades	17
Figura 2 –	O fluxo com as fases da visão computacional.....	18
Figura 3 –	Processo de captura da imagem	19
Figura 4 –	Separação das classes da máquina de vetores de suporte	24
Figura 5 –	Conjunto de dados não lineares	25
Figura 6 –	Histograma de gradientes orientados para detecção de pedestres ...	26
Figura 7 –	Passo a passo da técnica do histograma de gradientes orientados ..	27
Figura 8 –	Configuração dos pontos de referência facial e suas bases de dados	29
Figura 9 –	Pontos de referência detectando o olho aberto e fechado, e abaixo o <i>EAR</i> do olho encontrado em uma sequência de imagens, onde ocorre apenas um pestanejar	30
Figura 10 –	Sequência de passos realizado em cada imagem capturada para extrair apenas os pontos dos olhos	34
Figura 11 –	Sequência de imagens para detectar o pestanejar	36
Figura 12 –	Interface gráfica da aplicação e a exibição das notificações	38
Figura 13 –	Principais fatores que influenciaram em alguns erros do algoritmo	43
Tabela 1 –	Valores da média das variações de <i>EAR</i> e <i>EAR</i> mínimo dos voluntários	41
Tabela 2 –	Quantidade média de segundos para cada pestanejar de cada tarefa do experimento	49

LISTA DE GRÁFICOS

Gráfico 1 –	Percentual de acertos do algoritmo nos testes	40
Gráfico 2 –	Percentual de acertos do algoritmo na tarefa da leitura	42
Gráfico 3 –	Percentual de acertos do algoritmo na tarefa do vídeo	42
Gráfico 4 –	Quantidade de notificações emitidas na leitura e no vídeo	44
Gráfico 5 –	Percentual da quantidade de pestanejos que foram notificados na leitura	44
Gráfico 6 –	Percentual da quantidade de pestanejos que foram notificados no vídeo	45
Gráfico 7 –	Quantidade de pestanejos com e sem notificações na leitura de modo geral	46
Gráfico 8 –	Quantidade de pestanejos com e sem notificações no vídeo de modo geral	46
Gráfico 9 –	Quantidade de notificações dos voluntários que possuem adversidades	47
Gráfico 10 –	Quantidade de notificações dos voluntários que não possuem adversidades	48

LISTA DE ABREVIATURAS E SIGLAS

2D	Segunda Dimensão
3D	Terceira Dimensão
AAM	<i>Active Appearance Models</i>
ASM	<i>Active Shape Models</i>
CAA	Comunicação Alternativa e Aumentativa
CCD	<i>Charge Coupled Device</i>
CMOS	<i>Complementary Metal-Oxide Semiconductor</i>
EAR	<i>Eye Aspect Ratio</i>
HOG	<i>Histogram of Oriented Gradients</i>
LED	<i>Light Emitting Diode</i>
OpenCV	<i>Open Source Computer Vision Library</i>
RAM	<i>Random Access Memory</i>
SDM	<i>Supervised Descent Method</i>
SVM	<i>Support Vector Machine</i>
VDU	<i>Visual Display Unit</i>

SUMÁRIO

1	INTRODUÇÃO	12
1.1	Objetivos	13
1.1.1	<i>Objetivo geral</i>	13
1.1.2	<i>Objetivos específicos</i>	14
1.2	Justificativa	14
2	REVISÃO BIBLIOGRÁFICA E TRABALHOS RELACIONADOS	15
2.1	Síndrome visual do computador	15
2.2	Imagem digital	16
2.3	Visão computacional	17
2.4	Deteção facial e dos olhos	22
2.5	Máquina de vetores de suporte	23
2.6	Histograma de gradientes orientados	25
2.7	Pontos de referência facial	28
2.8	Pestanejar em tempo real com pontos de referência facial	30
2.9	Trabalhos relacionados	32
3	SISTEMA PARA DETECÇÃO DO PESTANEJAR UTILIZANDO VISÃO COMPUTACIONAL	33
4	RESULTADOS E DISCUSSÕES	40
5	CONCLUSÃO E TRABALHOS FUTUROS	50
5.1	Trabalhos futuros	50
	REFERÊNCIAS	52
	ANEXO A – TERMO DE AUTORIZAÇÃO DE IMAGEM	56

1 INTRODUÇÃO

Com o advento da internet, o uso dos dispositivos digitais, tais como computadores, *notebooks*, *smartphones* e *tablets*, aumentaram bastante. As pessoas utilizam dispositivos digitais por tempo prolongado, muitas vezes devido às exigências do trabalho, outras devido ao interesse por entretenimento, comunicação entre outros motivos.

O uso de dispositivos computacionais tende a manter o usuário concentrado com os olhos abertos expondo a superfície ocular e diminuindo a quantidade de pestanejamentos durante o uso do mesmo. Isto pode desidratar os olhos e provocar sintomas como desconforto ocular ou olho seco, bem como sintomas mais graves como visão desfocada ou visão dupla. Os indivíduos que contêm a síndrome visual do computador sofrem com alguns sintomas do conjunto de sintomas que compõem esta síndrome.

Comenta Charpe e Kaushik (2009), que a quantidade dos pestanejamentos na leitura de textos em papel é de vinte e duas vezes por minutos, enquanto esta quantidade diminui para sete vezes por minuto quando a leitura é feita nos computadores. Descreve Yan *et al.* (2008) que, durante uma conversa o ser humano pisca de 15-16 vezes por minuto, e que esta quantidade diminui para 5-6 vezes por minuto durante o uso do computador. Este autor inclui que, ocorre uma alteração na dinâmica da lágrima por causa do brilho da tela do computador, provocando significativamente a redução do pestanejar durante o uso do mesmo, devido a alta demanda visual e a grande concentração mental. Conclui também Patel *et al.* (1991) em sua pesquisa, que existe uma diminuição de 18,4 piscadas por minuto durante uma conversa, para 3,6 piscadas por minuto durante uma unidade de exibição virtual (do inglês, *Visual Display Unit - VDU*).

A visão computacional é a área que tenta simular a capacidade visual humana em um computador que busca descrever cenários e objetos nas imagens. Quando se refere a visão humana, não limita apenas à captura da imagem, conforme Backes e Sá Junior (2016), apesar de essa ser uma capacidade impressionante, ela é apenas o início de um processo muito vasto e rico. Para Gonzalez e Woods (2009), a visão é o sentido mais avançado do ser humano e que não é surpresa que as imagens exercem o papel mais importante da percepção humana.

Para que o computador consiga realizar a detecção de objetos existentes no ambiente de uma imagem digital, deve ser considerado algumas variações como o tamanho e orientação dos objetos, além de fatores que não dependem do objeto como a iluminação e ruídos na imagem. Além disso, o computador deve possuir um conhecimento antecipado do objeto em questão para poder classificá-lo. Existem várias técnicas para realizar a detecção facial com diferentes métodos, como métodos baseados em conhecimento ou aparências, que buscam encontrar o rosto e provê a localização na imagem, bem como técnicas que rastreiam características faciais e permite trabalhar individualmente com cada característica.

Sendo assim esta pesquisa tem por objetivo o desenvolvimento de uma aplicação que monitora o pestanejar do usuário em computadores e emite notificações de texto ou sonoro em uma faixa de tempo ideal para mantê-lo com os olhos lubrificados. Para isso, utilizou-se técnicas de detecção facial e pontos de referência facial para localizar os pontos que compõem a região dos olhos, onde é calculado o estado de abertura dos olhos. Em uma sequência de imagem verifica-se a alternância do estado dos olhos de aberto para fechado e aberto novamente constatando o pestanejar, com isso cronometrar o tempo que existir a ausência do pestanejar para emitir uma notificação ao usuário. Desse modo a aplicação percebe a falta do pestanejar durante o uso dos computadores e emite notificações ao usuário do dispositivo para que o movimento seja executado.

1.1 Objetivos

Com base na proposta desta pesquisa, tem-se os seguintes objetivos:

1.1.1 Objetivo Geral

O presente trabalho tem como objetivo desenvolver uma aplicação que auxilia os usuários de computadores a manter-se atento no pestanejar, afim de prevenir a síndrome visual do computador que é provocado pelo uso excessivo dos dispositivos. Sendo assim, a aplicação monitora o usuário em tempo real solicitando o pestanejar sempre que identificar a ausência do movimento. Para isto, utiliza-se técnicas de visão computacional.

1.1.2 Objetivos Específicos

- Realizar revisão bibliográfica sobre técnicas de detecção facial, olhos e movimentação do pestanejar;
- Aplicar técnicas de detecção facial, dos olhos e capturar o pestanejar;
- Implementar aplicação integrando as técnicas de visão computacional;
- Realizar testes do algoritmo de visão computacional;
- Validar os experimentos da aplicação.

1.2 Justificativa

Os avanços da tecnologia da informação trouxeram benefícios à sociedade, no entanto, resultou em uso contínuo e prolongado dos computadores provocando alterações na função visual, expondo a superfície ocular e diminuindo a quantidade do pestanejar, por isso o aparecimento de sintomas de fadiga visual, fotofobia, dificuldade em focar, visão dupla, entre outros, que caracterizam a síndrome visual do computador. O estudo propõe uma aplicação que auxilia os usuários a se manterem atentos ao pestanejar durante o uso dos computadores, sendo monitorados por uma câmera para captar o pestanejar e na ausência disto a aplicação emite uma notificação lembrando-o de realizar o movimento.

2 REVISÃO BIBLIOGRÁFICA E TRABALHOS RELACIONADOS

Neste capítulo são apresentados os principais conceitos utilizados no desenvolvimento dessa pesquisa.

2.1 Síndrome Visual do Computador

A síndrome visual do computador é definida por Neves e D'Araújo Filho (2019), como o uso excessivo do computador que resulta em sintomas visuais como a fadiga visual, irritação ocular, prurido, secura ocular, lacrimejamento, fotofobia, dor no pescoço e costas, entre outros sintomas. Acrescenta Gonçalves (2018), os seguintes sintomas causados por esta síndrome: dor de cabeça, dor nos olhos, dificuldade em focar e visão dupla. Algumas das causas conhecidas pela síndrome visual do computador classificadas como extrínsecas oculares são a diminuição do pestanejar e o aumento da exposição da superfície ocular (SÁ, 2016).

O olho humano é um sistema óptico complexo que funciona como uma lente que converge, sendo capaz o suficiente de focalizar os raios luminosos na retina. Existe uma lente composta por cinco camadas de células, chamada córnea, que permite a passagem da luz mantendo-se transparente devido a não possuir vasos sanguíneos e por manter líquidos que a lubrifica. A lubrificação externa da córnea é realizada pelo filme lacrimal, este que recobre a parte exposta do olho, é composto por três capas: uma capa mucosa diretamente em contato com a córnea que permite uma maior aderência, uma capa aquosa no intermédio com funções antimicrobiana, lubrificação, metabólica, entre outras e uma capa lipídica no exterior que reduz a evaporação da capa aquosa (ESTEPA, 2014).

O movimento das pálpebras de abrir e fechar dos olhos é responsável por propagar a lágrima por toda a região ocular. A lágrima é essencial para conservar os olhos lubrificados e a integridade da córnea, pois sua responsabilidade é nutrir, oxigenar, lubrificar, neutralizar e fornecer proteção antimicrobiana (MEDEIROS, 2017).

Na visão de Piccoli (2003), as transformações implantadas no ambiente de trabalho, principalmente os computadores, exige uma eficiência maior dos olhos e dos componentes do sistema nervoso relacionado. O estudo realizado por Sá (2016), afirma que os trabalhadores usuários de computadores, maioria com carga horária de

40 horas semanais, relataram que no trabalho sentem cansaço na vista e peso nos olhos, e em casa apenas o cansaço na vista. Após um exame oftalmológico as complicações predominantes entre os trabalhadores foram presbiopia e astigmatismo. Na pesquisa feita por Neves e D'Araújo Filho (2019) conclui que, foi constatado em 75,4% de seus participantes a síndrome visual do computador, e que os colírios lubrificantes são eficazes para redução dos sintomas.

Na pesquisa realizada por Gonçalves (2018), os sintomas que prevalecem nesta síndrome são as sensações de ardor e dor de cabeça, e pouco comum foram a visão dupla e halos à volta de objetos. Importante ressaltar também que nesta pesquisa os sintomas de dor de cabeça e visão desfocada de longe foram significativamente predominantes nos indivíduos do sexo feminino.

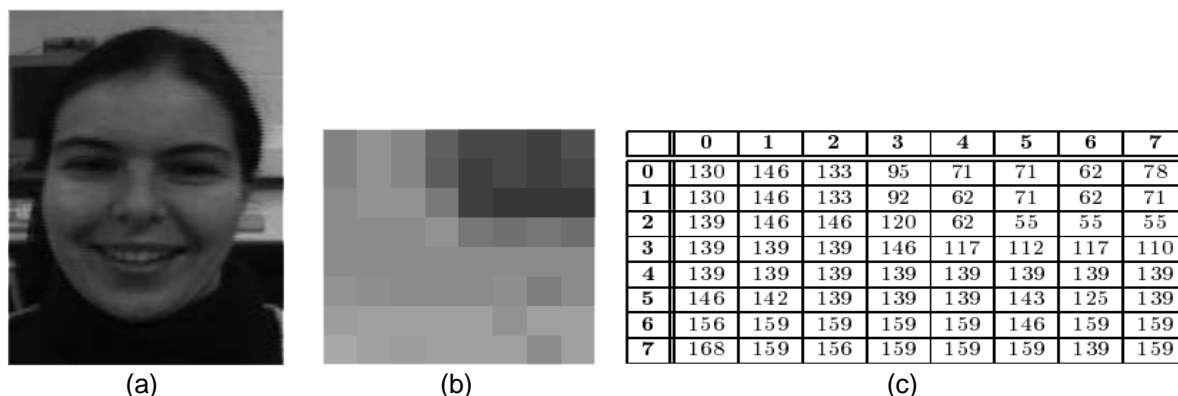
2.2 Imagem Digital

Antes de explicar os processos que acontecem nas imagens, é necessário entender sobre uma imagem digital. Portanto, uma imagem digital pode ser definida como uma função bidimensional, $f(x, y)$, onde os valores x e y são coordenadas no plano e f é o valor de intensidade situado em quaisquer combinação das coordenadas (x, y) . Quando x , y e os valores de f são quantidade finitas e discretas, chama-se de imagem digital (GONZALEZ; WOODS, 2009).

Definem Shapiro e Stockman (2000) que, uma imagem digital é composta por um número fixo de linhas e colunas formando pequenos blocos com valores quantizados com números pequenos, geralmente entre 0 e 255, que representa o brilho nos pontos da imagem.

Pode-se observar que na Figura 1, (a) representa uma imagem digital de uma face em tons de cinza, (b) é uma sub imagem retirada da imagem (a) contendo 8x8 pixels, e por fim, (c) representa os valores numéricos de intensidade da sub imagem (b).

Figura 1 - Representação de uma imagem digital com sua subimagem e seus respectivos valores de intensidades.



Fonte: Shapiro e Stockman (2000, p. 14, com adaptações)

Uma imagem possui um número finito de elementos, e que cada elemento tem sua localização e um valor específico de intensidade. Segundo Gonzalez e Woods (2009), esses elementos podem ser chamados de elementos pictóricos, elementos de imagem, pels e pixel. O termo comumente utilizado para representar os elementos da imagem digital é Pixel.

Uma imagem em escala de tons de cinza possui em cada pixel uma variação de valores de intensidade entre 0 (ausência total de luminosidade) a 255 (maior intensidade de luminosidade). Em uma imagem colorida pode ser vista como a associação de três imagens do mesmo tamanho, uma representando os pixels com tons de vermelho, a segunda com tons de verde e a terceira com tons de azul, onde a cor final é gerada pela soma dos valores dos pixels das três imagens (ALENCAR, 2015).

2.3 Visão Computacional

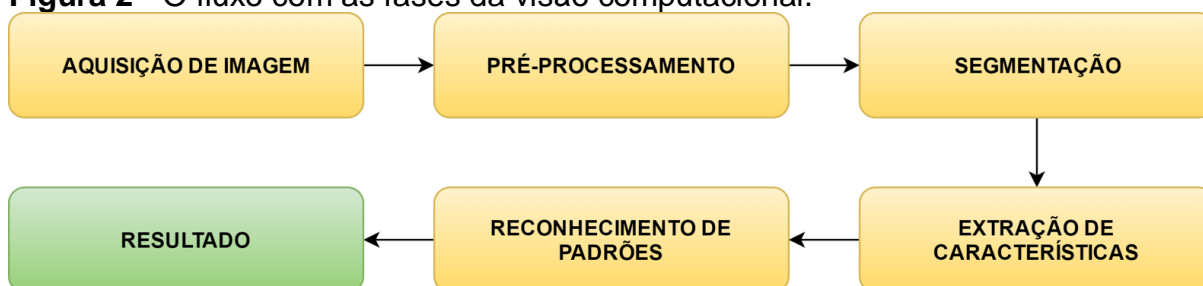
O ser humano consegue facilmente reconhecer as imagens e objetos, no entanto, é uma tarefa que tem demonstrado ser desafiadora em termos computacionais provocando grande interesse dos pesquisadores (FERREIRA, 2017). A visão computacional busca em uma imagem interpretar o cenário e objetos físicos ali existente. Afirma Alencar (2015), que esta é a área de estudo responsável por extrair dados informativos com base em uma imagem, geralmente utilizada para automatizar tarefas associadas à visão humana. O objetivo da visão computacional

consiste em tomar decisões pertinentes a partir de objetos reais e cenas de imagens capturadas (SHAPIRO; STOCKMAN, 2000).

Para que o computador consiga identificar os objetos reais presente em uma imagem é necessário um conhecimento prévio sobre esses objetos, segundo Shapiro e Stockman (2000), para tomar decisões sobre os objetos reais com base em uma imagem, quase sempre é necessária uma descrição ou modelos deles. Devido a este fato, de acordo com Ferreira (2017), a visão computacional tem como objetivo, definido por muitos pesquisadores, como a construção da descrição das cenas conseguidas em imagens.

Como descrito por Barelli (2018), um sistema de visão computacional normalmente possui um fluxo em comum com várias fases, como demonstrado na Figura 2, sendo elas: aquisição da imagem, pré-processamento, segmentação, extração de características e reconhecimento de padrões.

Figura 2 - O fluxo com as fases da visão computacional.

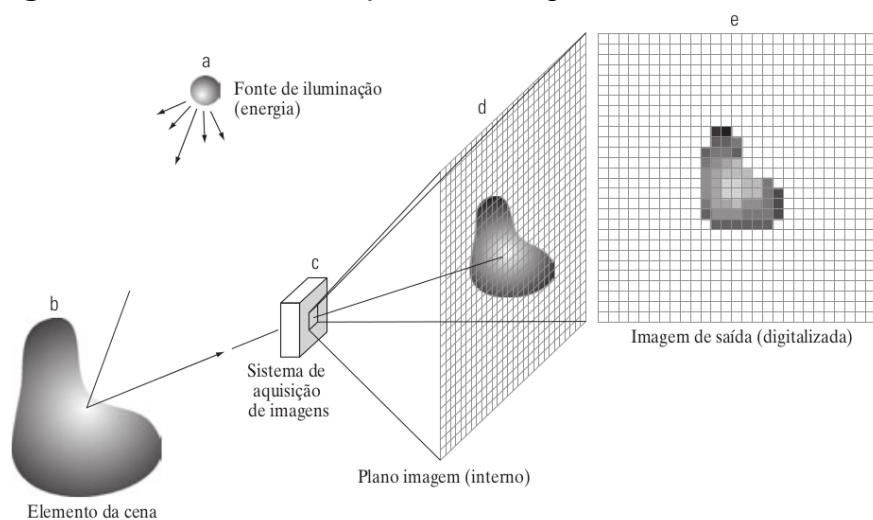


Fonte: Barelli (2018).

A aquisição gera uma imagem a partir da reflexão ou absorção da iluminação dos elementos da cena, isto é, a combinação de uma fonte de iluminação e os elementos presentes no cenário. A fonte de iluminação e a cena podem ser mais genéricas que uma situação habitual de uma cena tridimensional, por exemplo, a fonte de iluminação pode ser eletromagnética como sistema de radar e infravermelho, ou pode se originar por fontes tradicionais como a iluminação do computador, e de forma similar os elementos da cena podem ser objetos tradicionais, mas também podem ser moléculas ou um cérebro humano (GONZALEZ; WOODS, 2009).

Como ilustrado na Figura 3, o processo de captura da imagem, explica Gonzalez e Woods (2009) que, se a iluminação for uma luz, a entrada deste sistema é uma lente ótica que projeta o cenário sobre um plano focal da lente.

Figura 3 - Processo de captura da imagem.



Fonte: Gonzalez e Woods (2009, p. 32)

Os arranjos de sensores que coincidirem com o plano focal geram saídas equivalentes à integral da luz obtida em cada sensor. Uma varredura é feita nessas saídas pelos circuitos digitais e analógicos convertendo em um sinal analógico, que então é digitalizado por outro componente do sistema de aquisição gerando uma imagem digital como saída. A captura de imagens utilizando sensores matriciais é predominantemente encontrado nas câmeras digitais (GONZALEZ; WOODS, 2009).

Segundo Barelli (2018), os sensores de Dispositivo de Carga Acoplada (do inglês, *Charge Coupled Device* - *CCD*) e os Sensores Semicondutores de Metal-Óxido Complementar (do inglês, *Complementary Metal-Oxide Semiconductor* - *CMOS*) são as duas categorias de sensores eletrônicos mais utilizadas para o processo de geração da imagem digital. Os dispositivos que utilizam estes sensores são utilizados para capturar e digitalizar o ambiente gerando a imagem a qual será submetida a visão computacional.

O pré-processamento digital de imagem, conceitua Barelli (2018) que, é o processo responsável por realçar a imagem, simplificando a obtenção de suas informações. Este processo é sustentado por Gonzalez e Woods (2009), como um tipo de processo computacional de baixo nível, que envolve operações primitivas para reduzir ruídos, realce de contraste e o aguçamento de imagem. Geralmente com o propósito de aprimorar a qualidade ou retificar prováveis defeitos da imagem (ALENCAR, 2015). Esta fase pode apenas realizar as melhorias na imagem, bem como passar esta nova imagem adiante para as próximas fases.

A segmentação fragmenta uma imagem em regiões ou objetos que a constituem. Argumenta Shapiro e Stockman (2000), que é o particionamento de uma imagem em regiões que a cobrem, com o objetivo de fortalecer as regiões para representar áreas significativas nas imagens, e que essas regiões são grupos de pixels com borda e uma forma específica. O detalhe da fragmentação depende do problema a ser resolvido, segundo Gonzalez e Woods (2009), no momento em que os objetos ou regiões de interesse forem detectados a segmentação deve ser suspensa. Por exemplo, em uma imagem gerada por uma *selfie* poderia estar interessado em extrair o rosto da pessoa, ou um relógio que por ventura estivesse na imagem.

Afirma Shapiro e Stockman (2000) que, a segmentação possui dois objetivos. O primeiro é a decomposição da imagem em partes para análises posteriores, em casos simples pode-se extrair de maneira confiável as áreas que precisam ser analisadas de ambiente controlado, e em casos complexos a segmentação pode ser uma tarefa muito difícil e exigir um grande domínio de conhecimento. O segundo objetivo é realizar mudanças de representação organizando os pixels de forma que seja mais significativo ou mais eficiente para as análises.

Opina Gonzalez e Woods (2009), que um procedimento bem-sucedido de segmentação na imagem aumenta as chances de que os objetos sejam encontrados individualmente, entretanto, algoritmos de segmentação inconsistentes quase sempre garante falhas no processamento. Além disso, a segmentação de imagens não triviais é um dos trabalhos mais difíceis do processamento de imagem.

O processo de extração de características consiste na retirada de um conjunto de características importantes sobre um determinado objeto. Segundo Gonzalez e Woods (2009), este processo trabalha com a extração de características de interesse que resultam em alguma informação quantitativa ou possa ser aplicado para diferenciar classes de objetos. O termo característica refere-se ao fragmento de uma imagem com um determinado atributo em especial, como linhas, cantos, bordas, áreas de alto contraste de cor e regiões de textura (CORREIA, 2019). De acordo com Shapiro e Stockman (2000), o extrator de características retira informações relevantes para a classificação, esta extração de características é realizada normalmente pelo software. Por exemplo, para que uma aplicação consiga encontrar um smartphone existente em uma imagem, é necessário que o algoritmo possua uma coleção de características que define este objeto.

Surgiram várias técnicas para extrair características de imagem e as mais utilizadas são os algoritmos de detectores e descritores de características locais, onde o padrão de imagem que difere rapidamente de sua vizinhança é chamado de característica local, comumente representadas por pontos, bordas e pequenas regiões da imagem. Dessa forma, a busca realizada para localizar estas características que conseguem descrever bem as características buscadas é a função do detector de características locais. Assim, esses algoritmos buscam ser robustos em diversas transformações como pontos de vista, escala, rotação e iluminação (CORREIA, 2019).

E por fim, o reconhecimento de padrões é responsável por classificar o objeto em uma imagem com base em seu conjunto de atributos, Gonzalez e Woods (2009) afirma que, o reconhecimento é o processo de rotular o objeto com base em seus descritores, e padrões é o reconhecimento de regiões individuais da imagem. Conforme Shapiro e Stockman (2000), um sistema de reconhecimento deve possuir alguma memória do objeto que deseja reconhecer, e a construção dessa memória pode ser feita ensinando previsões de acordo com um grande número de amostras ou como o professor ensina o alfabeto. Prosseguindo com Gonzalez e Woods (2009), um padrão equivale a um arranjo de descritores (conjunto de características), um descritor apenas é comumente chamado na literatura de característica, e uma classe de padrões é um conjunto de padrões que partilham propriedades em comum.

De modo geral, na visão de Okada (2018), na bibliografia existe duas maneiras de reconhecer padrão, sendo elas, classificação supervisionada e não supervisionada. Na classificação supervisionada é rotulado o dado de entrada como pertencente a umas das classes pré-estabelecida no problema, e a não supervisionada o próprio algoritmo identifica a classe do dado, distinguindo e julgando com base nas características. Dessa maneira, o reconhecimento de padrão é uma tarefa de categorização, onde podem existir classes pré-definidas ou aprender com o decorrer do processo.

De acordo com Okada (2018), o reconhecimento de padrões possui três fases fundamentais:

- Aquisição de dados: Obtém coordenadas, imagens, vídeos ou características (nome, tamanho, cor, etc.) como dados de entrada. Nesta etapa, também é realizado a extração de características nas abordagens clássicas, onde seleciona os pontos de interesse pertencente nos dados de entrada.

- Representação de dados: Os dados são rotulados a categoria em que representa suas propriedades, diante disto, é realizado a classificação dos objetos.
- Tomada de decisão: Tem como função tomar decisões baseado nos resultados obtidos. Pode-se voltar as etapas com a finalidade de buscar novas características ou novos classificadores, em virtude de pouca efetividade nos resultados.

Importante enfatizar que não existe um padrão nas fases da visão computacional. Alguns autores denominam a fase de pré-processamento de imagens como processamento de imagens, e outros definem o processamento de imagens como um processo que envolve desde a aquisição da imagem até o reconhecimento dos objetos. A quantidade de fases também varia de acordo com os autores, além disso, alguma fase pode ser eliminada dependendo do contexto do problema.

A visão computacional possui relações com inúmeras outras áreas que colaboram com técnicas úteis, cita Ferreira (2017) que, como o processamento de imagens, computação gráfica, reconhecimento de padrões e inteligência artificial.

2.4 Detecção Facial e dos Olhos

O conceito de detecção de faces dada por Yang, Kriegman e Ahuja (2002), é que a partir de uma imagem, tem como objetivo de verificar se existe ou não rosto na imagem, caso exista, a localização na imagem deve ser fornecida e a extensão de cada face. Da mesma forma é realizado a detecção dos olhos, considerando que esta verificação pode ser realizada na região que foi identificado como uma face. Conforme Lopes (2003), detectar a face antes das características faciais individualmente, poupa muito trabalho de processamento, uma vez que, a maioria dos algoritmos baseiam a procura em toda a imagem, e desta forma diminui e limita uma nova região da imagem para realizar a busca das características faciais.

Existe uma série de fatores que influenciam na detecção e dificultam os algoritmos de visão computacional. De acordo com De Santana e Rocha (2015), os fatores que podem dificultar são os ruídos, variação de iluminação, expressões faciais, imagem de fundo, orientação da cabeça, obstrução da face ou sobreposição de faces. A orientação da cabeça, por exemplo, pode esconder partes importantes que o

algoritmo necessita capturar para realizar a detecção corretamente, assim como os demais fatores.

As detecções podem trazer dois tipos de erros, afirmado por Yang, Kriegman e Ahuja (2002), os falsos negativos nos quais os rostos existentes não são detectados e os falsos positivos nos quais uma região da imagem é identificada como face, mas nenhum rosto existe naquele local.

Para detectar as faces em uma única imagem, foi categorizada por Yang, Kriegman e Ahuja (2002), como quatro métodos de detecção, sendo eles: método baseado em conhecimento, em características invariantes, em modelo e em aparências.

1. Baseado em conhecimento. Implementam regras que extrai relações entre as características faciais, de acordo com o conhecimento humano de como é formado o rosto.
2. Baseado em características invariantes. Busca encontrar características estruturais que existem em faces mesmo na variação de pose, ponto de vista ou iluminação, e em seguida, utiliza essas características para localizar as faces.
3. Baseado em modelo. Possui armazenado vários padrões do padrão de uma face, para descrever um rosto como um todo ou as características faciais separadamente. A correlação entre a imagem de entrada com os padrões armazenados é computada para realizar a detecção.
4. Baseado em aparência. Em contraste com o método anterior, aprende a partir de um treinamento feito em uma coleção de imagens para coletar a variabilidade representativa da aparência facial. Este modelo treinado é utilizado para as detecções.

2.5 Máquina de Vetores de Suporte

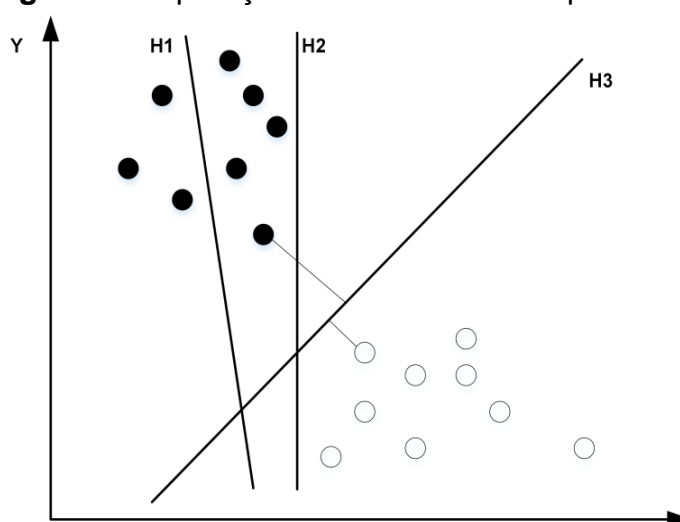
As Máquinas de Vetores de Suporte (do inglês, *Support Vector Machine - SVM*) é um método que visa a proposição de técnicas de aprendizado de máquina que procura maximizar a capacidade de generalização e minimizar o risco estrutural baseada na Teoria de Aprendizado Estatístico. Maximizar a capacidade de generalização é a eficiente classificação da máquina diante de um conjunto de

treinamento, e minimizar o risco estrutural é a probabilidade de classificações de padrões errados que ainda não foram mostrados à máquina (MENDES, 2017).

Fundamenta Platt (1998), que as máquinas de vetores de suporte é uma técnica de aprendizado de máquina, onde formam um hiperplano de tal forma que um conjunto de exemplos positivos e negativos são separados por uma margem máxima em sua forma mais simples e linear. Afirma Santiago (2017) que, os classificadores chamados de lineares são aqueles que dividem os dados por meio de um hiperplano. Um hiperplano é um subespaço que tem uma dimensão a menos que o espaço de configuração de ambiente, conforme Mendes (2017). A SVM procura construir um hiperplano de separação ótima, em que a distância entre duas classes é maximizada (DA COSTA, 2016).

Na Figura 4 observa-se diferentes hiperplanos que podem efetuar a separação entre as classes, entretanto, existe apenas um que máxima a distância entre as classes, e esse então é o hiperplano de separação ótimo.

Figura 4 - Separação das classes da máquina de vetores de suporte.



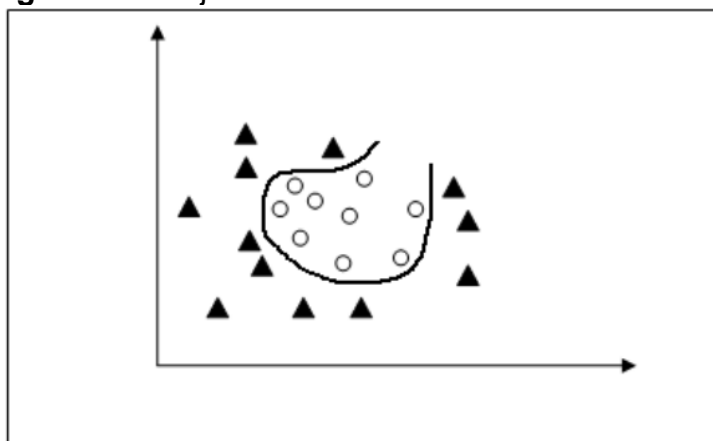
Fonte: Mendes (2017).

Conforme Ferreira (2017), devido às infinitas possibilidades de escolha para a margem que faz a separação dos exemplos, o objetivo é maximizar a distância dos exemplos positivos e negativos próximos da margem. Existem casos que não são possíveis separar linearmente de um conjunto de dados, portanto não há um hiperplano que consiga dividir todos os pontos negativos e positivos, neste caso, uma penalidade é aplicada no exemplo que falhar em se posicionar na margem correta.

As máquinas de vetores de suporte são modelos de aprendizado de máquina supervisionado, em que concedido uma coleção de exemplos de treinamento rotulados pertencentes a uma categoria será criado um modelo capaz de prever os rótulos de novos dados, transformando-o em um classificador linear binário não-probabilístico. Quando os dados não recebem os rótulos é necessária uma abordagem não supervisionada que procura, dos dados em grupo, determinar o agrupamento (*clustering*), para assim mapear novos dados para esses grupos (SANTIAGO, 2017).

A Figura 5 ilustra as máquinas de vetores de suportes não lineares que são usadas quando não existe a possibilidade do hiperplano dividir os dados de treinamento, portanto, as SVMs não lineares tratam os dados esquematizando o conjunto de treinamento do seu espaço original para um novo espaço chamado de espaço de características (SANTIAGO, 2017).

Figura 5 - Conjunto de dados não lineares.



Fonte: Santiago (2017).

De acordo com Mendes (2017), uma tarefa básica em aprendizado de máquina é a classificação de dados que parte da premissa de um novo ponto pertencer a uma das duas classes, e a finalidade é definir em qual classe este novo ponto será adicionado.

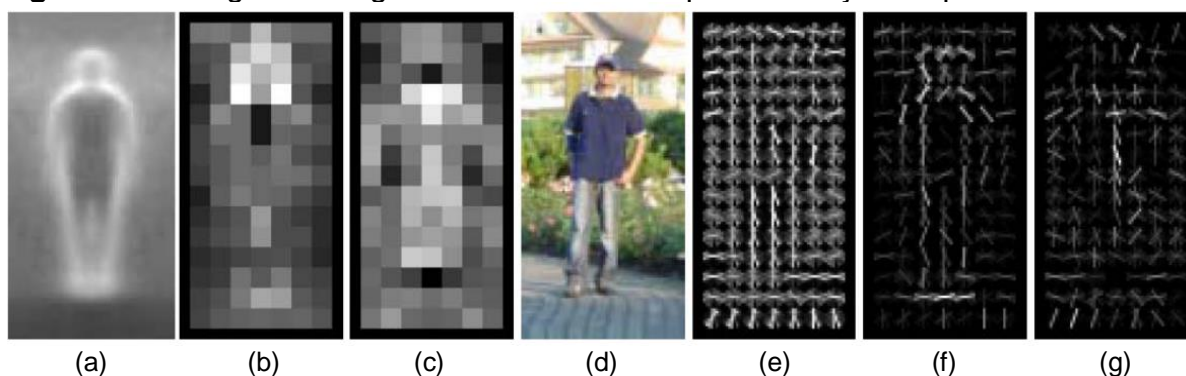
2.6 Histograma de Gradientes Orientados

O Histograma de Gradientes Orientados (do inglês, *Histogram of Oriented Gradients – HOG*), de acordo com Dalal e Triggs (2005), é a técnica que avalia os

histogramas locais bem normalizados de orientações de gradiente da imagem em uma grade densa. A ideia basicamente consiste que a aparência e a forma de um objeto podem regularmente ser descrito pela distribuição de intensidades do gradiente local ou às direções das bordas, mesmo sem conhecimento preciso do gradiente em questão ou posições de borda.

Como ilustra a Figura 6, o (a) indica, sobre os exemplos de treinamento, a imagem gradiente média, o (d) exibi uma imagem de teste como entrada, enquanto o (e) mostra a associação dos descritores *HOG*. As imagens (b) e (c) demonstram os pesos positivos e negativos em formato de diagramas, nesse meio tempo que (f) e (g) exibi, correspondente a imagem de entrada, os retornos de *HOG* ponderados. O (f) corresponde ao retorno ponderado pelos pesos positivos e o (g) ao retorno ponderado pelos pesos negativos (SZELISKI, 2010).

Figura 6 - Histograma de gradientes orientados para detecção de pedestres.



Fonte: Szeliski (2010).

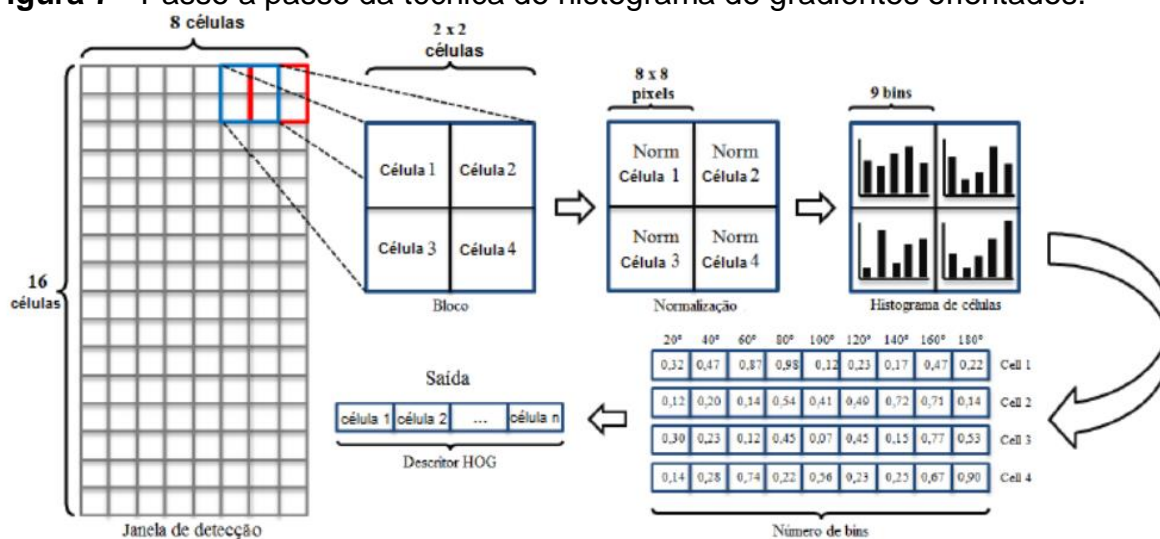
Para Szeliski (2010), este método utiliza um conjunto de descritores de sobreposição de histograma de gradientes orientados nutrido em uma máquina de vetores de suporte. Em outras palavras, como descrito por Do Vale Manuel *et al.* (2016), o histograma de gradientes orientados é uma técnica para detectar e extrair características de uma imagem e a máquina de vetores de suporte é um classificador para essas características.

Na prática a janela da imagem é dividida em regiões pequenas, nomeadas de células, que armazenam um diagrama de orientações de inclinação ou orientações de bordas locais sobre os pixels que a compõe. É feito a normalização do contraste dos histogramas locais antes de utilizá-las para melhorar a invariância da iluminação, sombreamento, entre outros, e para isso concentra-se uma medida do histograma

local sobre regiões espaciais relativamente grandes e usando os resultados para normalizar todas as células do bloco, refere-se esses blocos de descritores normalizados como descritores do histograma de gradientes orientados (DALAL; TRIGGS, 2005).

Explica Júnior (2016), que esta técnica representa em partes distintas da imagem a variação das orientações dos gradientes, de modo que, não varie com a iluminação e que em cada pixel da imagem a magnitude dos gradientes são computados em diferentes orientações e reunidos em blocos, posteriormente, as barras (*bins*) do histograma, as células e os blocos são normalizados. Com uma imagem em escala de cinza para extrair os descritores *HOG*, duas matrizes de mesma dimensão são geradas contendo a magnitude do gradiente e a orientação dos pixels, esses valores são encontrados a partir da derivada (I_x, I_y) em cada pixel. A sequência de passos desta técnica é ilustrada na Figura 7.

Figura 7 - Passo a passo da técnica do histograma de gradientes orientados.



Fonte: Júnior (2016, com adaptações)

Para extrair o histograma de gradientes orientados de uma imagem em escala de cinza, foi definido em quatro etapas por Júnior (2016), (1) sobre a imagem de gradientes deslocar a janela de detecção, (2) são extraídos bloco de 2x2 células a partir da detecção, (3) cada célula que pertence ao bloco possui 8x8 pixels, e (4) cada histograma de células é composto por nove barras representadas em uma faixa de 0 à 180 graus, e o vetor de características final é composto pelo conjunto de todos os

histogramas dos gradientes de cada região. Este vetor é enviado à uma máquina de vetores de suporte linear para classificar se é ou não o objeto.

2.7 Pontos de Referência Facial

O *Landmarking* é o procedimento de mapeamento dos pontos de interesse (marcos ou *landmarks*) em imagens 2D ou modelos 3D de um objeto ou sujeito (WAGNER, 2017). Esses pontos, também conhecido como pontos de referência, podem ser marcados manualmente, porém existem formas automáticas que realizam este processo. Um algoritmo de pontos de referência facial, segundo AbdAlmageed *et al.* (2016), obtém uma imagem de rosto e estima uma quantidade de n pontos chaves pré-definidos que observa as características faciais como os olhos, boca, nariz, etc. Dependendo da quantidade de pontos chaves que foram definido, os pontos de referência podem se classificar em pontos de referência esparsos como na coleção de pontos multifuncional de cinco pontos, e pontos de referência densos como na coleção de pontos de 68 pontos.

Conforme Tong *et al.* (2012), o modelo de conhecimento geométrico ou pontos de referência realiza o alinhamento de face supervisionado anotando um conjunto de pontos de referência em cada imagem que descreve as principais características faciais para ser submetido ao treinamento. Os detectores de pontos de referência modernos são treinados em “*in-the-wild datasets*”¹, deste modo, possui robustez à variação de iluminação, várias expressões faciais e as rotações moderado da cabeça (CECH; SOUKUPOVA, 2016).

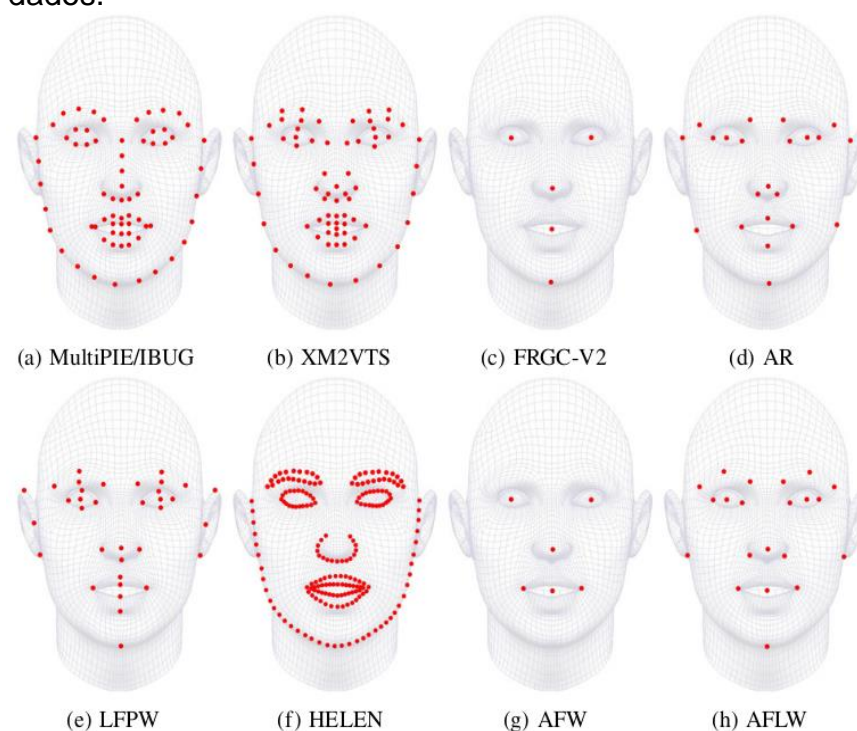
Observa Sagonas *et al.* (2016), que muitas pesquisas científicas estão sendo realizadas na visão computacional para localizar pontos de referência facial utilizando modelos de visuais deformáveis, onde foi proposto várias metodologias para demonstrar grande precisão e eficiência. Essas metodologias podem ser divididas em duas categorias: generativo e discriminativo. Os métodos generativos têm em vista encontrar, gerado pelo modelo, os parâmetros que maximizam a probabilidade da imagem do teste, incluem Modelos de Aparência Ativo (do inglês, *Active Appearance Models - AAM*). E os métodos discriminativos podem ser decompostos nos que utilizam funções de mapa de resposta discriminativa, como Modelos de Forma Ativa

¹ Imagens obtidas sob condições totalmente irrestritas (SAGONAS *et al.*, 2016, p. 5).

(do inglês, *Active Shape Models - ASM*), e aqueles que aprendem uma cascata de funções de regressões, como exemplo a técnica descentralizado supervisionado (do inglês, *Supervised Descent Method - SDM*).

Na literatura existem vários métodos propostos para localizar os pontos de referência sob configurações semisupervisionadas ou pouco supervisionadas. Porém, Sagonas *et al.* (2016) afirma que, estes métodos possuem duas grandes limitações. A primeira delas é solicitar informações adicionais sobre as imagens inserida, por exemplo, apontar os pixels pertencente a área da face manualmente em cada uma das imagens para o treinamento, ou a orientação das faces nas imagens de treinamento. E a segunda limitação é que só foi aplicado em imagens com condições controladas. Além disso a anotação manual de grandes bases de dados é um processo que demanda muita carga horária de trabalho e um especialista treinado, além de que, em alguns casos as anotações são imprecisas por causa da fadiga e falta de concentração, destacando a necessidade de uma ferramenta de anotação semiautomática. A Figura 8 ilustra as configurações e quantidade dos pontos de referência facial e seus bancos de dados associados, os quais foram utilizados em uma proposta de um método de anotações semiautomático.

Figura 8 - Configuração dos pontos de referência facial e suas bases de dados.



Fonte: Sagonas *et al.* (2016).

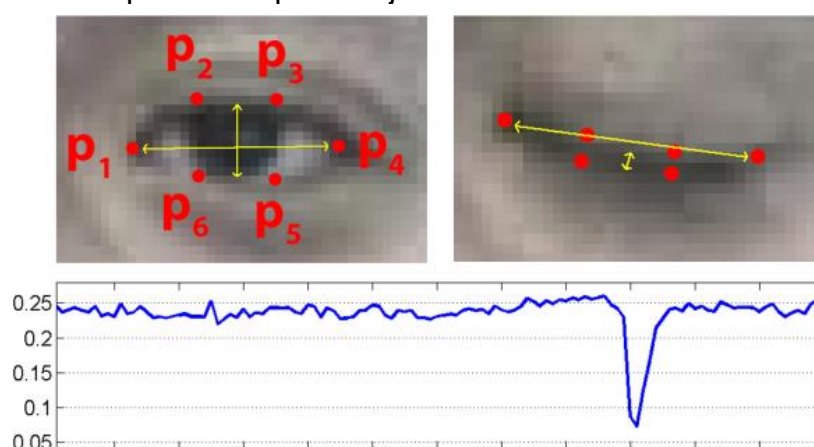
Esta técnica semiautomática, proposta por Sagonas (2016), para marcação dos pontos de referência facial tem como intuito de anotar novamente os grandes bancos de dados faciais para superar, das anotações originais, os principais problemas. Eles alegam em seu trabalho que durante os últimos anos foram realizados muitos progressos, no entanto, o problema da exatidão do alinhamento facial está distante de ser solucionado e que existe lugar para adicionar melhorias.

2.8 Pestanejar em Tempo Real com Pontos de Referência Facial

O algoritmo proposto por Cech e Soukupova (2016), é simples, porém eficiente para detectar o pestanejar utilizando detecção de pontos de referência facial. A técnica deles consiste em uma quantidade escalar única, chamado de relação de aspecto do olho (do inglês, *Eye Aspect Ratio* - *EAR*), derivado dos pontos de referência que representa o grau de abertura do olho. Para identificar o pestanejar em uma sequência de imagens com as estimativas de abertura dos olhos foi treinado um classificador com a máquina de vetor de suporte com exemplos de padrões de piscou e não piscou.

A Figura 9 demonstra o olho com os pontos de referência facial, denominados por p_1 , p_2 , p_3 , p_4 , p_5 e p_6 , no estado em que o olho está aberto e logo ao lado no estado do olho fechado, e abaixo a variação dos valores do *EAR* em uma sequência de imagens, onde a queda do *EAR* e a volta do mesmo representa o pestanejar.

Figura 9 - Pontos de referência detectando o olho aberto e fechado, e abaixo o *EAR* do olho encontrado em uma sequência de imagens, onde ocorre apenas um pestanejar.



Fonte: Cech e Soukupova (2016).

De posse dos pontos de referência detectado na imagem, deriva-se a relação de aspecto do olho (EAR) que é utilizado como estimativa do estado de abertura dos olhos (CECH; SOUKUPOVA, 2016). O valor do EAR é obtido a partir da resolução da Equação (1).

$$EAR = \frac{||p2 - p6|| + ||p3 - p5||}{2 ||p1 - p4||} \quad (1)$$

Segundo Cech e Soukupova (2016), a relação de aspecto do olho é computada entre altura e largura do olho, onde $p1$ a $p6$ são localizações dos pontos de referência facial. O valor do EAR é constante quando o olho está aberto e se aproxima de zero enquanto fecha o olho, e este valor também possui uma pequena variância entre os indivíduos e é totalmente invariável para uma escala uniforme da imagem e rotação no plano da face.

O pestanejar, que dura cerca de 100-400 milissegundos, é o fechamento e reabertura de um olho rapidamente, este processo é diferente para cada indivíduo podendo diferir na velocidade do fechamento e abertura, no grau de apertar o olho e na duração intermediária. Como o pestanejar é realizado pelos dois olhos de forma sincronizada, calcula-se a média dos EAR de ambos os olhos (CECH; SOUKUPOVA, 2016).

As contribuições deste trabalho, afirma Cech e Soukupova (2016), é a capacidade de dois detectores de pontos de referência diferenciar de forma confiável entre os estados aberto e fechado dos olhos que é demonstrado quantitativamente em uma coleção de dados desafiadores “*in-the-wild*”, e para diversas resoluções de imagem do rosto, bem como um algoritmo novo de detecção de pestanejar em tempo real que integra com um detector de pontos de referência, e assim um classificador é proposto.

Exemplifica Cech e Soukupova (2016) que, o método proposto por Sukno (2009), baseado em modelos de forma ativa, possui um tempo de processamento de aproximadamente cinco segundos por quadro de segmentação, e a indicação da abertura do olho é normalizado por estatísticas estimadas por uma sequência mais longa, sendo assim, ele é utilizável apenas para processamento off-line.

2.9 Trabalhos Relacionados

Nos últimos anos tem-se desenvolvido muitas pesquisas na área de visão computacional provocando uma grande evolução. As técnicas produzidas por estes trabalhos estão sendo aplicadas na criação de aplicações que auxilia o ser humano. Para melhor entendimento do âmbito geral deste trabalho foi descrito a seguir alguns trabalhos relacionados que utilizaram algumas das técnicas que este trabalho também faz uso, bem como por se tratar da prevenção da síndrome visual do computador. Importante salientar que os resultados das buscas realizadas não tiveram trabalhos imediatamente relacionados ao que foi abordado nesta pesquisa.

O trabalho de Silva *et al.* (2019), de título *Lógica Fuzzy Aplicada ao Controle de Intensidade da Luz*, regula a intensidade da luz transmitida dos visores digitais de forma automática, conseqüentemente uma suavização adequada para o ser humano que lida com monitores todos os dias. Esta automatização ocorre pela utilização da lógica de Fuzzy quem aumenta as possibilidades de controle automático a emissão de luz adequada a luminosidade local, e o arduino cuida de regular a intensidade enviada ao diodo emissor de luz (do inglês, *Light Emitting Diode – LED*). Os autores concluem que 9% da energia total consumida pelos dispositivos são de responsabilidade das telas, e que o projeto reduz em média 4% deste consumo.

O projeto de Sardinha *et al.* (2016), de título *Um Componente para Detecção de Piscada de Olhos Utilizando a Interação Homem Computador*, desenvolve um componente que identifica a piscada dos olhos do usuário e emite um comando para o computador, o intuito desta aplicação é buscar um meio de comunicação, de baixo custo e acessível, para pessoas com Síndrome do Encarceramento. Esta ferramenta concede a criação de aplicações de Comunicação Alternativa e Aumentativa (CAA) sem a necessidade do conhecimento relacionado ao processamento de imagem.

O trabalho de Ferrassini (2014), de título *Sistema de Detecção de Sonolência*, por meio de Visão Computacional, implementa uma visão computacional que detecta o estado de sonolência do motorista em um veículo com o propósito de prevenir acidentes de trânsito. O algoritmo conta com a técnica de Viola e Jones para detecção da face e dos olhos, posteriormente é aplicado apenas na região dos olhos, a técnica *thresholding* para identificar em qual estado o olho se encontra (aberto ou fechado) em um ambiente com iluminação controlado. Foram realizados testes bem-sucedidos, de 95% de acerto, do estado dos olhos do usuário.

3 SISTEMA PARA DETECÇÃO DO PESTANEJAR UTILIZANDO VISÃO COMPUTACIONAL

Estudos foram realizados sobre as técnicas de processamento de imagem para detecção de faces e olhos, bem como extração do pestanejar em tempo real, que servirão de base para o desenvolvimento da aplicação. Este trabalho é uma aplicação que busca auxiliar os usuários de computadores e notebooks a ficarem atentos aos pestanejos durante o uso desses dispositivos para auxiliar na prevenção da síndrome visual do computador.

Os recursos de hardware para desenvolvimento, testes e experimentos foram um notebook com um Intel® Core™ i3, da quinta geração de 2.00 GHz, memória *RAM* (*Random Access Memory*) de 8GB, com webcam integrada e o sistema operacional Linux Mint 19.01.

O sistema foi desenvolvido para computadores de mesa conectados a um webcam ou notebooks que utilizam o sistema operacional *Linux* ou *Windows*. O público alvo para o sistema são pessoas em seu local de trabalho ou ambientes de entretenimento que exijam um intenso uso dos computadores, como os órgãos públicos, empresas privadas de desenvolvimento, pessoas que utilizam com excesso o computador para seu divertimento, entre outros.

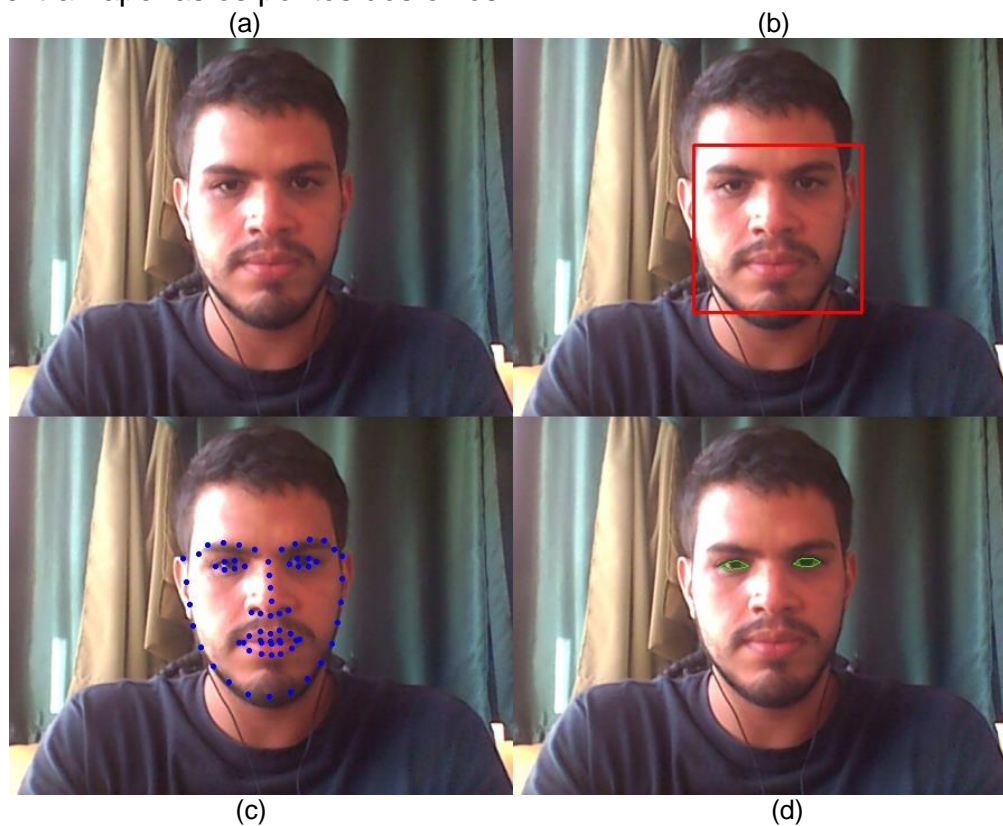
No desenvolvimento da aplicação foi analisado a quantidade de imagens consecutivas que o usuário permanece de olhos fechados. Essa variável é responsável por armazenar a quantidade de imagens em que os olhos estão fechados para relacionar com as imagens seguintes, uma vez que, se essa quantidade for acima de 0 significa dizer que a próxima imagem em estado dos olhos aberto será a ocorrência de um pestanejar. Após os testes realizados foi determinado que com pelo menos uma das imagens no estado dos olhos fechados, a próxima imagem na sequência que estiver no estado dos olhos abertos contabilizará o pestanejar.

No desenvolvimento da aplicação foram testados os valores de 0,20 a 0,30 para o *EAR* mínimo, e estipulou-se inicialmente o valor 0,20, conforme definido por Cech e Soukupova (2016). Entretanto, este valor teve que ser modificado por resultar em contagens errôneas dos pestanejos devido a variação do *EAR* de alguns voluntários serem baixo, valores estes que foram próximo ao *EAR* mínimo estabelecido. Em outras palavras, o *EAR* mínimo variou de voluntário para voluntário e foi necessário avaliar nos experimentos o valor do *EAR* mínimo ideal para cada voluntário.

Para o funcionamento da aplicação é necessário extrair três informações, sendo elas, detectar a face na imagem que foi capturada, posteriormente identificar os pontos de referência facial, e por fim, calcular o *EAR* nos pontos de referência dos olhos para distinguir do estado dos olhos de aberto e fechado. De posse dessas informações a próxima etapa é relacionar o *EAR* de cada imagem com as imagens seguintes para identificar o pestanejar nas imagens que são obtidos da câmera.

Para o desenvolvimento da aplicação foi utilizado a linguagem de programação *Python*². Dessa forma a aplicação seguiu os seguintes passos: capturar a imagem, detectar a face, identificar os pontos de referência facial e extrair apenas os pontos dos olhos, realizar o cálculo do *EAR* para constatar o pestanejar, e por fim, cronometrar o tempo em que o usuário não pestaneja para a aplicação exibir as notificações solicitando o pestanejar. Os quatro primeiros passos, são ilustrados na Figura 10, com os processamentos necessários para cada imagem individual.

Figura 10 – Sequência de passos realizado em cada imagem capturada para extrair apenas os pontos dos olhos.



Fonte: Elaborado pelo autor, 2019.

² Linguagem de programação orientada a objetos e interpretada. Ela possui uma sintaxe clara de fácil aprendizado e dispõe de módulos que aumenta a produtividade no desenvolvimento de aplicações (BORGES; DA SILVA; BEZERRA, 2016).

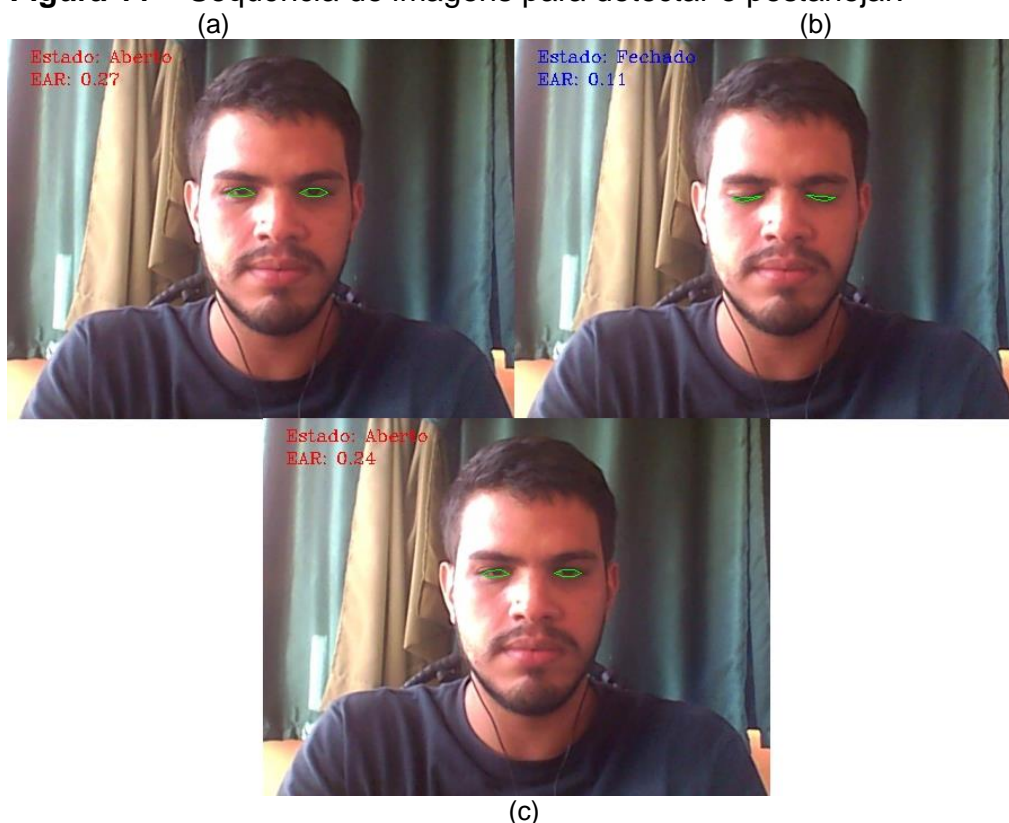
O primeiro passo, demonstrado na Figura 10 (a), é ativar a câmera do dispositivo em que o usuário esteja utilizando. A câmera é responsável por monitorar o usuário e capturar as imagens onde será aplicado o processamento de imagem, ou seja, este sensor é responsável por fornecer a imagem para os processos de visão computacional detectar o pestanejar. Para isso foi utilizado a biblioteca *imutils*, especificamente o pacote de vídeo, para ativar a câmera e capturar as imagens a partir da câmera do dispositivo definindo as dimensões da imagem com 450x337.

O segundo passo, ilustrado na Figura 10 (b), é a detecção facial que será utilizada para obter, da imagem de vídeo capturada em tempo real, apenas a parte que representa o rosto de uma pessoa. Para isto, a técnica usada foi o histograma de gradientes orientados com a biblioteca *Dlib*³ e o detector de faces frontais para detectar e extrair a localização da face. Ele percorre cada imagem em busca de uma face, ao encontrá-la retornam a sua localização.

O terceiro passo, apresentado na Figura 10 (c), é utilizar a região da face do passo anterior para localizar os pontos de referência facial, desse modo, pode-se mapear separadamente as principais características faciais. O importante para a aplicação é apenas a localização dos pontos dos olhos onde será realizado os cálculos para distinguir o estado de aberto ou fechado, portanto, apenas estes pontos foram trabalhados, logo este é o quarto passo como visto na Figura 10 (d). Para isto, também foi utilizado a biblioteca *Dlib* com o preditor de 68 pontos de referência facial.

Para identificar o estado dos olhos a partir dos pontos foi utilizado a derivação da relação de aspecto do olho (*EAR*) proposto por Cech e Soukupova (2016), que é o cálculo entre a altura e largura do olho de acordo com pontos de referência facial. Para constatar que os olhos estão no estado aberto ou fechado foi definido um valor mínimo de *EAR*. Logo, enquanto que o *EAR* variar em uma faixa de valores acima do valor mínimo representará o estado dos olhos aberto, se o *EAR* do usuário cair para abaixo do valor mínimo, então neste momento o estado dos olhos será fechado. Se na sequência de imagens capturadas ocorrer o estado dos olhos aberto, posteriormente o estado dos olhos fechado e novamente voltar para o estado dos olhos aberto, naturalmente essa sequência linear de imagens relacionadas representa o pestanejar, como demonstra a Figura 11.

³ Uma biblioteca que possui algoritmos de aprendizado de máquina e visão computacional que são usadas para desenvolver aplicações solucionando problemas do mundo real (SAMPALIO *et al*, 2018).

Figura 11 – Sequência de imagens para detectar o pestanejar.

Fonte: Elaborado pelo autor, 2019.

O quinto passo é verificar em uma sequência de imagens, como explicado anteriormente, a ocorrência com os dois olhos sincronizados a mudança do estado aberto para o fechado e para o aberto novamente comprovando o pestanejar. Para exemplificação da Figura 11, estabelecendo o *EAR* mínimo de 0,20 a Figura 11 (a) representa o estado dos olhos aberto com o valor do *EAR* de 0,27, estando acima do mínimo definido, em seguida a Figura 11 (b) ilustra o estado dos olhos fechado com o *EAR* de 0,11, ficando abaixo do mínimo, e por fim, a Figura 11 (c) retorna o estado dos olhos para aberto novamente com o valor do *EAR* de 0,24, voltando o valor para acima do mínimo, dessa maneira, caracterizando o pestanejar do usuário para a visão computacional.

E por último, o sexto passo é cronometrar o pestanejar do usuário para avisá-lo quando deve pestanejar, levando em consideração a quantidade necessária de pestanejos por minuto definido pela pesquisa de Yan *et al.* (2008). Se após o tempo estabelecido o usuário não pestanejar, então a aplicação será responsável por emitir uma notificação solicitando a movimentação.

O cálculo feito para definir a quantidade mínima de segundos para cada pestanejar do usuário está demonstrado na Equação (2), onde (S) é os segundos correspondente a 1 minuto e (Q) é a quantidade de pestanejos por minuto considerando a pesquisa de Yan *et al.* (2008). Logo, o quociente resultando (T) é o tempo mínimo em segundos que o usuário pode passar sem pestanejar. Exemplificando, a aplicação aguardará 1 pestanejar a cada 4 segundos ou 2 a cada 8 segundos e assim por diante. Caso a opção selecionada for 8 segundos, o sistema exibirá a notificação a cada 8 segundos e permanecerá visível ou tocando para o usuário até ele pestanejar um total de 2 vezes dentro deste tempo.

$$T = \frac{S}{Q} \quad (2)$$

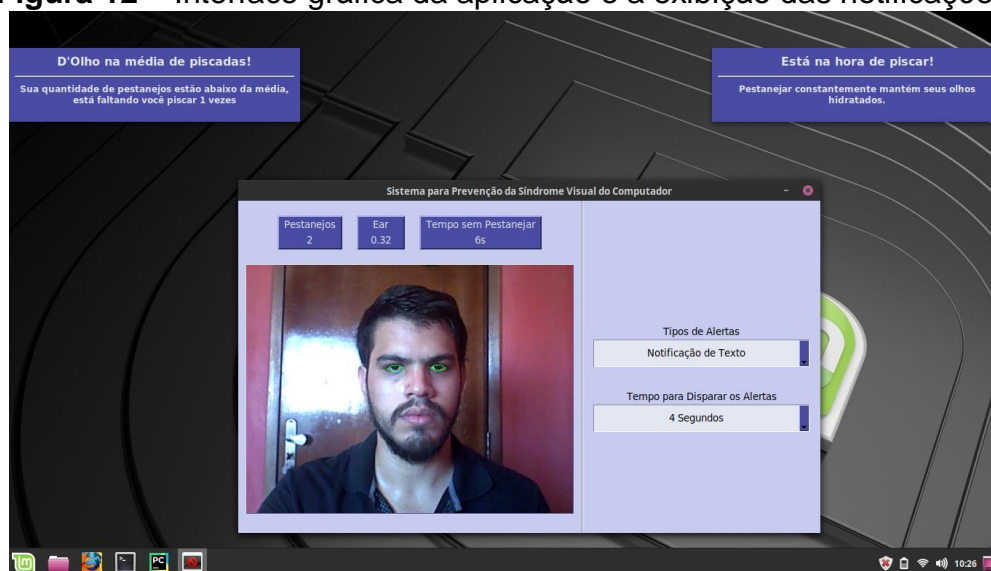
Por conseguinte, está notificação lembrará ao usuário de pestanejar caso o mesmo tenha ultrapassado a quantidade mínima do tempo sem realizar o movimento a quantidade de vezes necessária. Existe também uma outra notificação que avisa ao usuário se ele não está pestanejando a quantidade necessária em relação ao tempo de uso da aplicação. Para isso como ilustra a Equação (3), foi calculado a quantidade de pestanejos faltantes onde, (TT) contabiliza o tempo total até o presente momento da execução da aplicação, (T) é o tempo mínimo em segundos que o usuário pode passar sem pestanejar e (QA) é a quantidade total e atual de pestanejos do usuário, então está equação resulta na quantidade de pestanejos faltantes ou a mais (P) do usuário na duração do uso da aplicação.

$$P = \frac{TT}{T} - QA \quad (3)$$

O quociente entre (TT) e (T) resulta na quantidade de pestanejos que o usuário deveria ter realizado no tempo total de uso da aplicação, e a subtração deste quociente com (QA) retornará os pestanejos faltante ou a mais. A equação anterior pode resultar as seguintes opções: se o resultado for o valor zero, isto representa que o usuário está pestanejando no prazo ideal para manter os olhos hidratados; se o resultado for um valor positivo significa que o usuário não está pestanejando dentro do prazo, então está é a quantidade de pestanejos faltantes; e por fim, se o resultado

for um valor negativo indica que o usuário está pestanejando uma quantidade de vezes acima do estipulado no prazo, então está é a quantidade de pestanejos do usuário a mais que o necessário. A Figura 12 ilustra a interface gráfica da aplicação.

Figura 12 – Interface gráfica da aplicação e a exibição das notificações.



Fonte: Elaborado pelo autor, 2019.

A interface gráfica do sistema possui no lado esquerdo a exibição do vídeo com a imagem processada e os contornos dos olhos desenhado. Para a geração dos contornos dos olhos foi usado biblioteca OpenCV⁴ (*Open Source Computer Vision Library*). Logo acima da sequência de imagens é o campo de exibição de informações sobre o vídeo, com três informações importantes, sendo elas a quantidade total de pestanejos do usuário, o valor do *EAR* gerado por cada imagem e a quantidade de tempo em segundos que o usuário ficou sem pestanejar.

Ao lado direito é disponibilizado duas opções de configurações dando flexibilidade ao usuário alternar entre os tipos de notificações e o tempo mínimo sem pestanejar para que a aplicação exiba a notificação. A primeira configuração da aplicação conta com as opções de notificação por mensagem que surge no topo direito da tela solicitando o pestanejar ou a opção de notificação sonora, ambas permanecem executando até o usuário pestanejar. A segunda configuração do tempo mínimo que o sistema aguarda o pestanejar do usuário para então exibir a notificação possui as

⁴ Uma biblioteca voltada à visão computacional que se concentra no processamento de imagens e vídeo. Além disso ela é uma plataforma cruzada significando ser possível sua implementação e operação em diferentes sistemas operacionais (PAJANKAR, 2015).

opções de 4, 8 ou 12 segundos com a quantidade de 1, 2 ou 3 pestanejos respectivamente. Esse valor foi estabelecido de acordo com a pesquisa de Yan *et al.* (2008) onde afirma que o ser humano pestaneja de 15-16 vezes por minuto durante uma conversa, dessa maneira, pode-se estimar de 1 pestanejo a cada 4 segundos.

A alteração do tempo tem o intuito de melhorar a experiência do usuário com a aplicação, uma vez que, dependendo da atividade que estejam exercendo, talvez não seja do agrado a exposição da notificação em um prazo muito curto de tempo. Assim como a notificação sonora que fornece a alternativa de lembrar ao usuário sem que o oculte informações na tela.

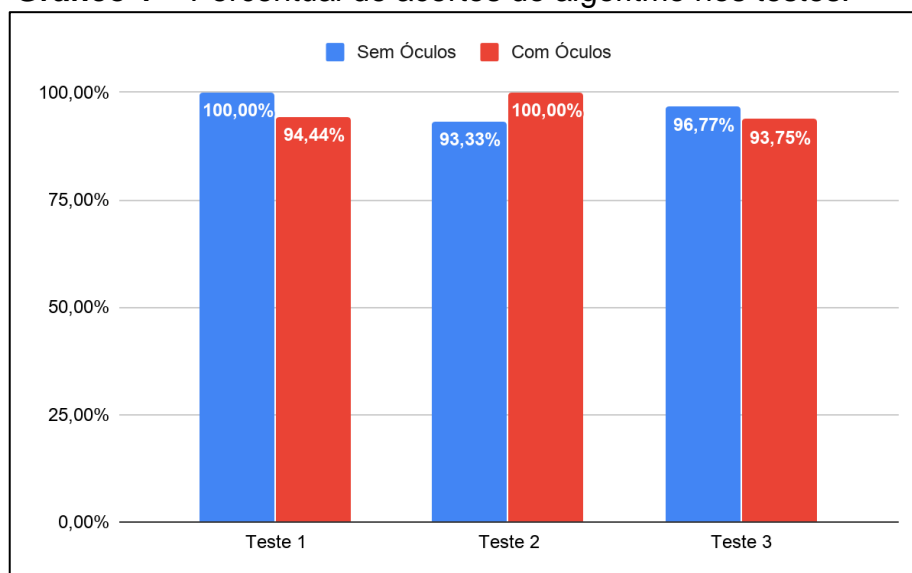
Para a execução eficiente da visão computacional nos testes e experimentos foi necessário organizar o ambiente. Uma boa iluminação foi fundamental para que a reflexão da luz nos voluntários resultasse em uma imagem nítida. Solicitou-se aos voluntários uma distância mínima do notebook de 41cm. E também foi requisitado que seus olhos ficassem alinhado ao topo do dispositivo, em virtude disso foi utilizado um suporte de notebook para realizar ajustes de altura do dispositivo para o voluntário.

4 RESULTADOS E DISCUSSÕES

Esta pesquisa surgiu após observar que o uso prolongado dos dispositivos está diminuindo o piscar dos usuários contribuindo para a síndrome visual do computador. Dessa maneira, foi desenvolvida uma aplicação para computadores com webcam e notebooks para monitorar os olhos do usuário e notificar o mesmo para piscar, e se está piscando a quantidade necessária durante o tempo de utilização do dispositivo.

Com intuito de validar a precisão do algoritmo de visão computacional foi realizado 3 testes com óculos e 3 testes sem óculos no vídeo em tempo real com duração de 1 minuto cada. Estes dois cenários foram considerados e analisados, pois notou-se que o uso dos óculos apresentava uma maior variância no *EAR*. O Gráfico 1 demonstra que tanto nos testes que foram usados os óculos quanto aos que não foram utilizados tiveram resultados satisfatórios com um percentual médio entre os testes de 96,38% de acerto.

Gráfico 1 – Percentual de acertos do algoritmo nos testes.



Fonte: Elaborado pelo autor, 2019.

Posteriormente para analisar a eficiência da aplicação foi efetuado um experimento com 10 voluntários, onde foi explicado para cada um deles o objetivo da aplicação, bem como os passos que tinham que ser executados. Desse modo, todos os voluntários efetuavam o experimento conhecendo o intuito principal da aplicação.

Os experimentos consistiam na execução da aplicação, ao mesmo tempo que os voluntários liam uma notícia e posteriormente assistiam um vídeo definido pelo pesquisador. Os conteúdos determinados foram a notícia “WhatsApp admite envio ilegal de mensagens durante as eleições de 2018” publicado no Olhar Digital e o vídeo “5 Invenções geniais que deveriam existir em todo lugar” no YouTube pelo canal de Fatos Desconhecidos. O tempo consumido nos experimentos da leitura variou por voluntário tendo uma média de 4 minutos entre todos os voluntários, enquanto que o vídeo possui um tempo definido no próprio conteúdo de 4 minutos e 23 segundos.

Nos primeiros experimentos alguns voluntários tiveram sua variação de *EAR* próximo ao valor mínimo definido no começo, desse jeito, esta variação, que se encontrava no estado aberto dos olhos, geravam valores que indicavam para a aplicação que o voluntário estava alternando no estado aberto e fechado, ou seja, estes valores variavam rapidamente abaixo e acima do mínimo contabilizando o pestanejar, sendo que o voluntário continuava com os olhos totalmente abertos, portanto o valor da variação do *EAR* do voluntário não pode estar próximo ao *EAR* mínimo. Por isso, na execução dos experimentos foi determinado um valor de *EAR* mínimo ideal para cada voluntário, como ilustra a Tabela 1.

Tabela 1 – Valores da média das variações de *EAR* e *EAR* mínimo dos voluntários.

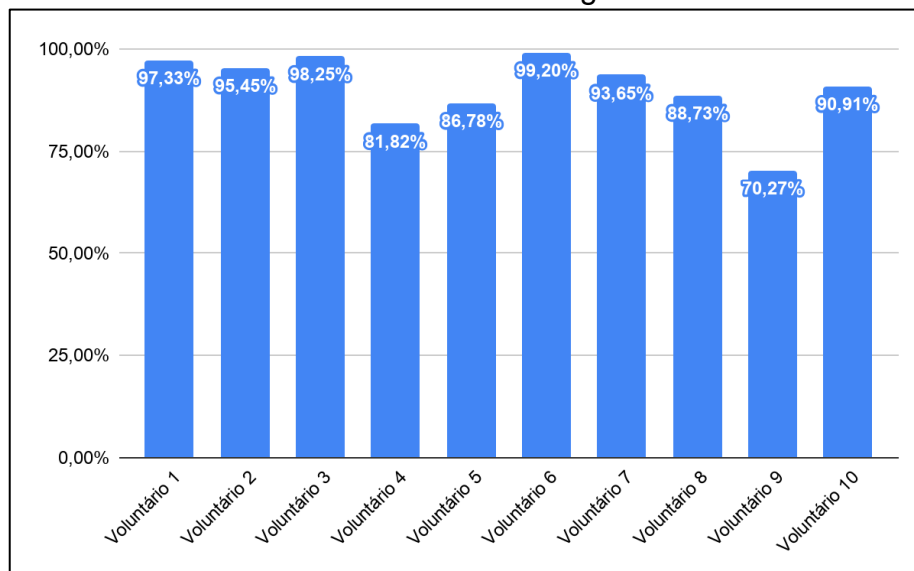
Voluntários	Média da variação do <i>EAR</i>	<i>EAR</i> Mínimo
Voluntário 4	0,20	0,12
Voluntários 6 e 7	0,21 e 0,22	0,13
Voluntários 5 e 9	0,21 e 0,23	0,15
Voluntário 8	0,23	0,17
Voluntários 1, 2, 3 e 10	0,28 a 0,30	0,20

Fonte: Elaborado pelo autor, 2019.

Foi estabelecido o *EAR* mínimo para cada voluntário individual considerando a variação de seus *EARs*, neste momento testes foram realizados em busca deste valor que melhor se adequasse ao voluntário. Na análise dos experimentos foram contabilizados a quantidade de pestanejos manualmente em 65.970 imagens que foram armazenadas na execução das tarefas. Está contagem manual foi comparada a contagem do algoritmo de forma a identificar o percentual de acerto. Essa abordagem foi considerada devido aos resultados satisfatórios dos testes, e também

devido ao tempo que não foi suficiente para efetuar a análise detalhada de todas as imagens geradas nos experimentos. O Gráfico 2 e o Gráfico 3 demonstra a eficiência em percentual dos acertos considerando essa abordagem.

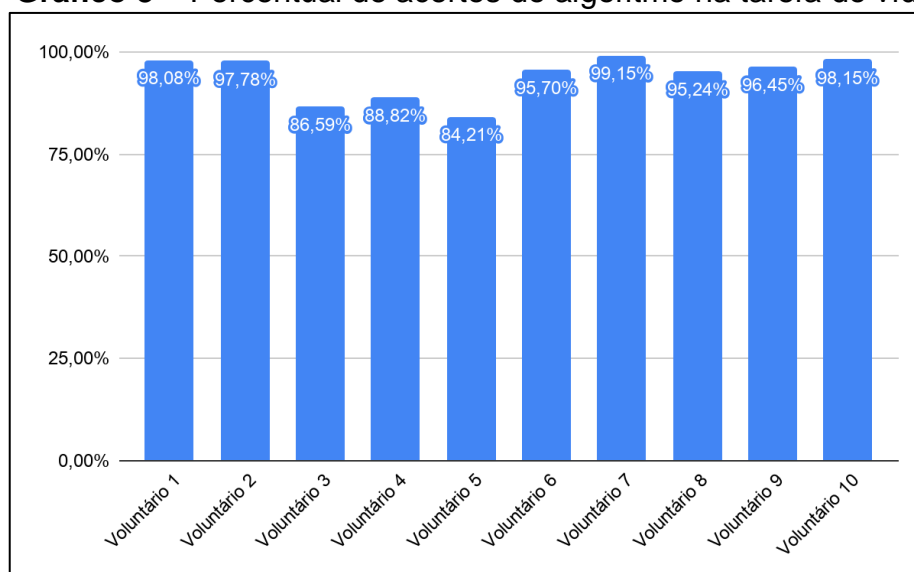
Gráfico 2 – Percentual de acertos do algoritmo na tarefa da leitura.



Fonte: Elaborado pelo autor, 2019.

Observa-se nos gráficos que os resultados do algoritmo foram satisfatórios com cada voluntário mantendo o percentual de acertos acima de 70%, tanto na leitura quanto no vídeo, concebendo um percentual médio entre os voluntários de 90,24% na tarefa da leitura e 94,02% na tarefa do vídeo.

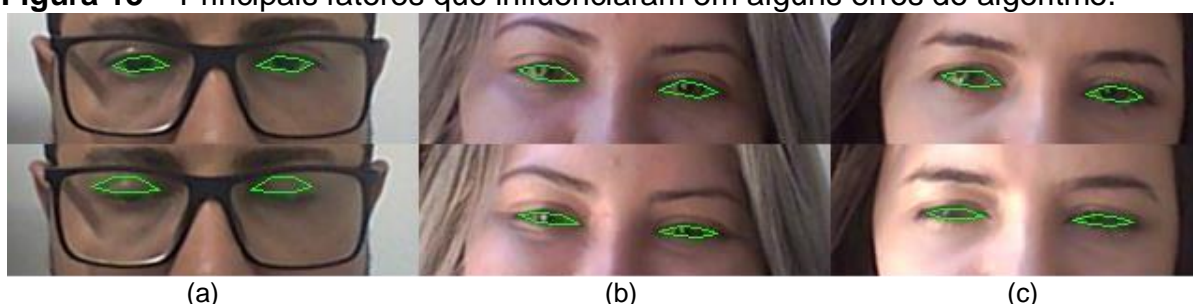
Gráfico 3 – Percentual de acertos do algoritmo na tarefa do vídeo.



Fonte: Elaborado pelo autor, 2019.

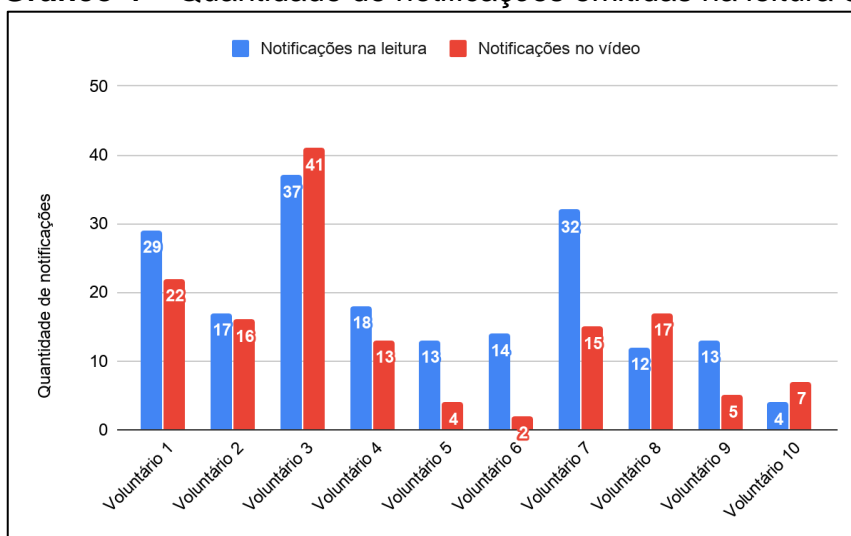
Na execução dos experimentos foram constatados alguns fatores que confundiram a visão computacional, como ilustra a Figura 13. O primeiro desses fatores, Figura 13 (a), foi a detecção da região dos olhos de forma errônea, dessa maneira, mesmo que os voluntários fechassem os olhos o algoritmo identificava o estado dos olhos em aberto e não contabilizava o pestanejar. Outro fator, Figura 13 (b), deve aos sorrisos enquanto os voluntários realizavam a tarefa, isso comprimia os olhos e resultava na queda do *EAR* indicando a visão computacional que o estado dos olhos foram para fechado, e após o sorriso o *EAR* subia novamente retornando ao seu estado dos olhos para aberto significando o pestanejar, contudo, os voluntários não pestanejaram de fato, apenas deixavam os olhos entre abertos. O último fator, Figura 13 (c), foi que alguns voluntários em um dado momento apertava os olhos para focar na tarefa, resultando na detecção errônea do pestanejar igualmente ao caso do sorriso. Esses breves momentos foram os falsos positivos (sorrisos e apertos dos olhos) e falsos negativos (a região dos olhos eram detectadas de forma errôneas).

Figura 13 – Principais fatores que influenciaram em alguns erros do algoritmo.



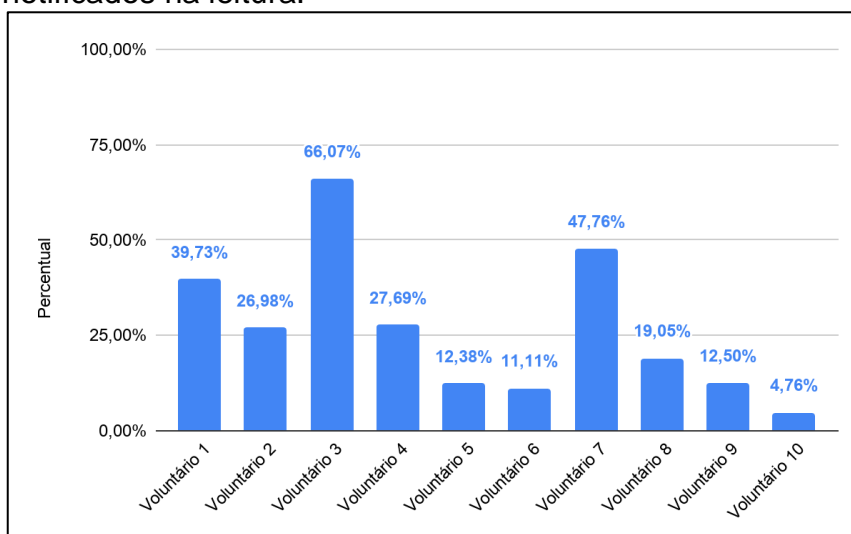
Fonte: Elaborado pelo autor, 2019.

Em relação a precisão da aplicação, notou-se que, mesmo que os voluntários soubessem do intuito do sistema, o foco na tarefa do experimento fazia eles esquecerem de pestanejar, dessa forma, a aplicação emitia a notificação solicitando o movimento. O Gráfico 4 ilustra a quantidade de notificações exibidas à cada voluntário nos experimentos.

Gráfico 4 – Quantidade de notificações emitidas na leitura e no vídeo.

Fonte: Elaborado pelo autor, 2019.

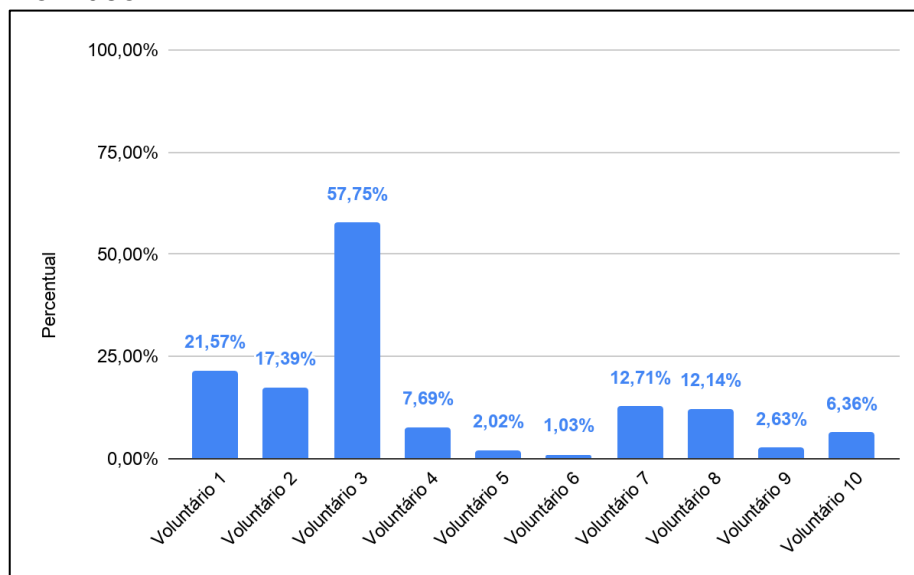
A emissão das notificações ocorreu com mais frequência na tarefa da leitura se comparado com as notificações da tarefa do vídeo. Os voluntários afirmaram que a leitura é estática e exige um foco que os deixam mais suscetível ao esquecimento do pestanejar, enquanto que o vídeo é mais dinâmico com imagens e áudios que exigem menos da concentração em relação à leitura. O Gráfico 5 demonstra o percentual da quantidade de pestanejos que foram necessários serem notificados na tarefa da leitura.

Gráfico 5 – Percentual da quantidade de pestanejos que foram notificados na leitura.

Fonte: Elaborado pelo autor, 2019.

Nota-se que aplicação interviu com frequência os voluntários nos experimentos, com um percentual em média de 26,80% dos pestanejos sendo lembrados pela aplicação. O Gráfico 6 demonstra o percentual da quantidade de pestanejos que foram necessários serem notificados na tarefa do vídeo.

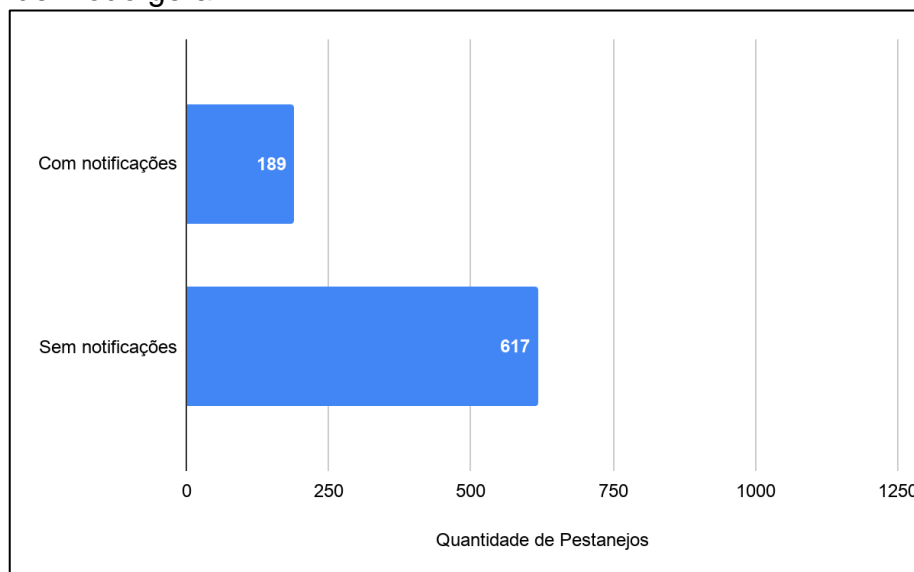
Gráfico 6 – Percentual da quantidade de pestanejos que foram notificados no vídeo.



Fonte: Elaborado pelo autor, 2019.

Em relação a leitura, as notificações para lembrar o pestanejar na tarefa do vídeo diminuiu, contando com um percentual em média de 14,13%. Neste caso pode constatar que na tarefa do vídeo a maioria dos voluntários permaneceram pestanejando constantemente sem a intervenção da aplicação, enquanto que na leitura esta frequência de pestanejos foi menor. Contudo, mesmo com a dinâmica da tarefa do vídeo foi necessário a interceptação da aplicação na tarefa solicitando o pestanejar. Importante destacar que a aplicação para o voluntário 3, tanto na leitura quanto no vídeo, foi essencial. O Gráfico 7 expressa na tarefa da leitura a quantidade de pestanejos de modo geral que foram notificados, bem como os que não tiveram essa necessidade.

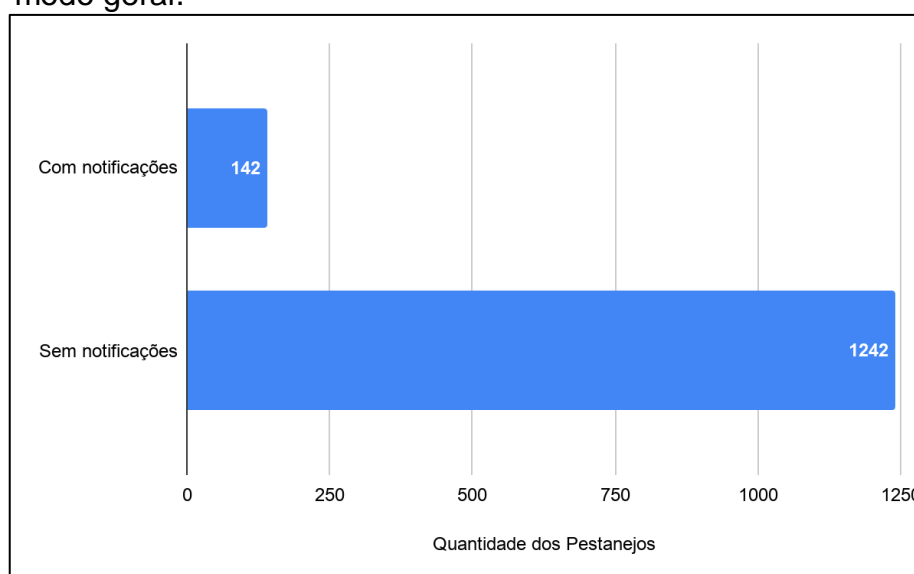
Gráfico 7 – Quantidade de pestanejos com e sem notificações na leitura de modo geral.



Fonte: Elaborado pelo autor, 2019.

A aplicação teve que notificar aos voluntários 189 vezes solicitando o pestanejar, isso contabiliza um percentual de 23,4% em relação ao total de pestanejos executados na leitura, a porcentagem restante representa os pestanejos que não tiveram esta necessidade. O Gráfico 8 ilustra na tarefa do vídeo a quantidade de pestanejos que foram e não notificados de modo geral.

Gráfico 8 – Quantidade de pestanejos com e sem notificações no vídeo de modo geral.

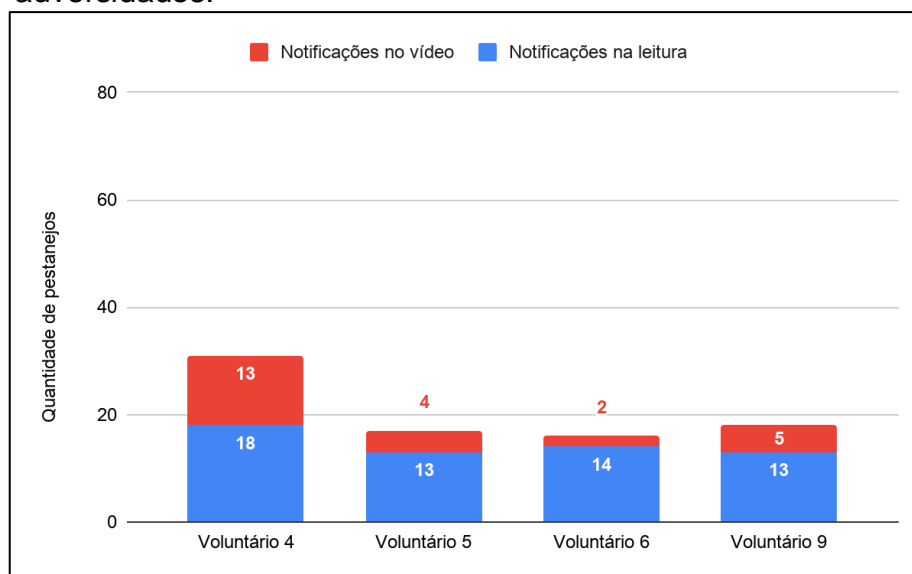


Fonte: Elaborado pelo autor, 2019.

Na tarefa do vídeo a aplicação requisitou aos voluntários que pestanejassem uma quantidade de 142 vezes, este valor representa um percentual de 10,3% em relação ao total pestanejos executados no vídeo, enquanto que o percentual restante não teve esta necessidade.

Analisando os dados da emissão das notificações, repara-se que a tarefa da leitura em relação ao vídeo exibiu mais notificações solicitando o pestanejar, em contrapartida, a tarefa do vídeo dobrou a quantidade de pestanejos que não precisou ser notificado. O Gráfico 9 demonstra separadamente a quantidade de notificações emitidas pedindo o pestanejar dos voluntários que afirmaram possuir algum tipo de adversidade, como astigmatismo ou olho seco, que faziam eles pestanejarem constantemente.

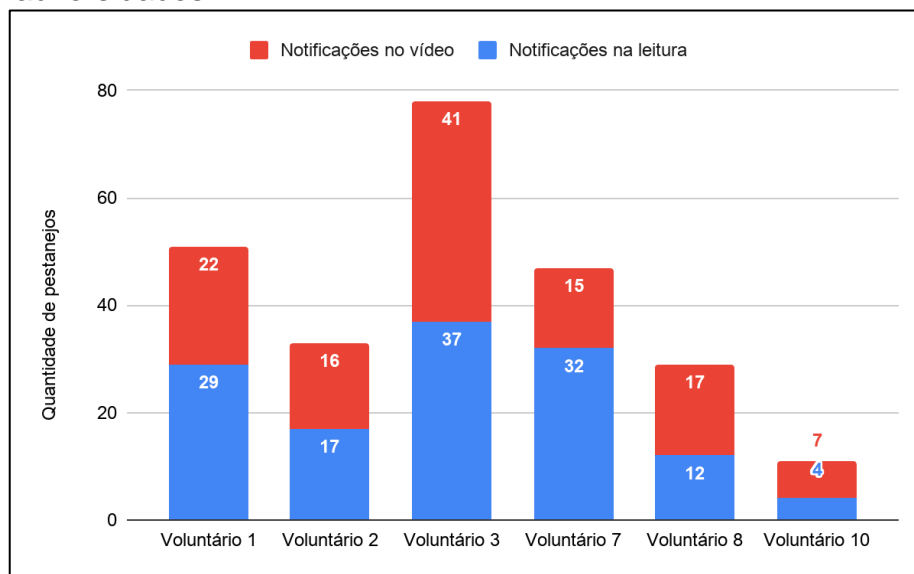
Gráfico 9 – Quantidade de notificações dos voluntários que possuem adversidades.



Fonte: Elaborado pelo autor, 2019.

Pode-se observar que a quantidade de exibições de notificações que lembram os voluntários de pestanejar foi baixa para aqueles que declararam alguma adversidade em comparação com os voluntários que não as possuíam, conforme ilustra o Gráfico 10.

Gráfico 10 – Quantidade de notificações dos voluntários que não possuem adversidades.



Fonte: Elaborado pelo autor, 2019.

Comparando um percentual médio das notificações que foram necessários lembrar o pestanejar entre os voluntários com e sem adversidades, tem-se um percentual de 76,21% das notificações que foram dos voluntários sem adversidades na realização de suas tarefas, enquanto que os que possuíam tiveram um percentual de 23,79%. Mesmo que a adversidade mantivesse os voluntários pestanejando frequentemente, ainda assim tiveram pestanejos esquecidos e lembrados pela intervenção da aplicação. No entanto, os voluntários que não possuíam adversidade tiveram a participação da aplicação constantemente recordando-os de pestanejar. A Tabela 2 demonstra quantos segundos demorou para cada pestanejo dos voluntários considerando o tempo total e a quantidade total de pestanejos durante o uso da aplicação, ela também ilustra um comparativo entre a contagem manual e a automática de modo a reparar a eficiência obtida pela aplicação.

Tabela 2 – Quantidade média de segundos para cada pestanejar de cada tarefa do experimento.

Voluntários	Quantidade média de segundos para cada pestanejar			
	Pela Aplicação		Manualmente	
	Leitura	Vídeo	Leitura	Vídeo
Voluntário 1	3,55	2,79	3,45	2,74
Voluntário 2	3,35	3,10	3,20	3,17
Voluntário 3	4,14	4,06	4,07	3,51
Voluntário 4	3,23	1,68	3,82	1,87
Voluntário 5	2,25	1,40	2,17	1,62
Voluntário 6	2,46	1,45	2,48	1,52
Voluntário 7	4,16	2,39	4,43	2,41
Voluntário 8	2,89	1,99	2,56	1,90
Voluntário 9	2,32	1,45	1,63	1,40
Voluntário 10	2,18	2,54	2,38	2,58

Fonte: Elaborado pelo autor, 2019.

Segundo Yan *et al.* (2008), uma pessoa pestaneja normalmente de 15 à 16 vezes por minuto, dessa forma, pode-se estimar de 1 pestanejo a cada 4 segundos. De acordo com a tabela anterior, a avaliação da aplicação comunica que 8 entre os 10 voluntários ficaram dentro do prazo, significando que eles preservaram seus olhos lubrificados durante a execução dos experimentos. Por outro lado, 2 entre os 10 voluntários ultrapassaram mesmo que pouco o prazo estimado e isso representa a não lubrificação ideal dos olhos.

Alguns voluntários pestanejaram em uma quantidade de tempo pequeno que indica um número maior de pestanejos por minuto, neste caso, vários fatores podem ser considerados. Primeiramente esses voluntários respondiam as notificações da aplicação com mais pestanejos do que o solicitado, ocorrendo um acúmulo e resultando em uma quantidade maior de pestanejos. Alguns voluntários afirmaram também que o uso da aplicação apresenta uma pressão mantendo-os atento e pestanejando em um determinado momento. E o último fator é que alguns voluntários afirmaram possuir astigmatismo ou olho seco que faziam eles pestanejarem frequentemente.

5 CONCLUSÃO E TRABALHOS FUTUROS

A aplicação desenvolvida neste trabalho para auxiliar na prevenção da síndrome visual do computador com visão computacional alcançou um percentual de acerto de 90,24% na tarefa da leitura e 94,02% na tarefa do vídeo totalizando uma média de 92,13% em conformidade com os dados analisados dos experimentos dos 10 voluntários. Notificando a eles para pestanejar no prazo estabelecido de acordo com a pesquisa de Yan *et al.* (2008) mantendo os olhos lubrificados, tendo como público alvo os usuários que permanecem uma carga horária elevada utilizando os computadores e notebooks.

O módulo de visão computacional obteve um percentual de acerto de 96,38% em relação aos falsos positivos e falsos negativos nos dois cenários estabelecidos nos testes, possuindo confiabilidade em seu uso para a detecção do pestanejar em tempo real.

As opções de configurações dos tipos de notificações entre texto ou sonoro, bem como a quantidade mínima do tempo que o usuário pode passar sem pestanejar fornecem aos usuários uma melhor experiência no uso da aplicação ajustando as notificações conforme suas preferências. Estas opções possui o intuito de flexibilizar a aplicação para diferentes atividades, como operar pacotes office, utilizar redes sociais ou jogos eletrônicos, nos computadores proporcionando uma agradável utilização da aplicação nestes dispositivos.

A aplicação mostrou-se eficiente não apenas para os usuários que planejam serem lembrados dos pestanejos mantendo seus olhos hidratados, mas também para realizar pesquisas sobre a síndrome visual do computador otimizando os processos manuais com opções automáticas, possibilitando o armazenamento das imagens dos testes efetuados, bem como relatório de informações relevantes para posteriores análises.

5.1 Trabalhos futuros

Comparar técnicas de processamento de imagem tendo em vista identificar qual agregaria melhor a aplicação. Além disso, trabalhar detalhadamente com as informações geradas pelos experimentos buscando novos resultados em pesquisa.

Ainda como trabalhos futuros, tem-se a versão *mobile* que abarca muitos dos usuários que ficam constantemente conectados na internet com o intuito principal de entretenimento. Vale salientar que o desenvolvimento desta versão foi iniciado, mas devido ao prazo de entrega não foi possível finalizar e incluir na pesquisa.

REFERÊNCIAS

ABDALMAGEED, Wael *et al.* Face Recognition Using Deep Multi-pose Representations. In: **2016 IEEE Winter Conference on Applications of Computer Vision (WACV)**. IEEE, 2016. p. 1-9.

ALENCAR, Francisco Eduardo Silva. **Desenvolvimento de um Sistema para Classificação Automática de Imagens Dermatoscópicas para Dispositivos Móveis**. 2015. Dissertação (Mestrado em Ciência da Computação) - Universidade do Estado do Rio Grande do Norte, Universidade Federal Rural do Semi-Árido, Rio Grande do Norte, 2015.

BACKES, André Ricardo; SÁ JUNIOR, Jarbas Joaci de Mesquita. **Introdução à visão computacional usando Matlab**. Alta Books Editora, 2016.

BARELLI, Felipe. **Introdução à Visão Computacional: Uma abordagem prática com Python e OpenCV**. Espírito Santo: Casa do Código, 2018. E-book.

BORGES, Rafael Assis; DA SILVA, Sebastião Simão; BEZERRA, Alysso Aldrin Barreto. Estudo de Trelças Planas e Espaciais Utilizando a Linguagem de Programação Python e o Software VTK. **Revista Interdisciplinar de Pesquisa em Engenharia**, v. 2, n. 27, 2016.

CECH, J.; SOUKUPOVA, T. Real-Time Eye Blink Detection using Facial Landmarks. **21st Comput. Vis. Winter Work.**, 2016.

CHARPE, Namrata Arora; KAUSHIK, Vandana. Computer Vision Syndrome (CVS): Recognition and Control in Software Professionals. **Journal of Human Ecology**, Rajasthan, v. 28, n. 1, p. 67-69, 2009.

CORREIA, Alana de Santana. **Uma Investigação a Respeito de Sistemas de Atenção Visual como Extratores de Características Locais**. 2019. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal de Sergipe, Sergipe, 2019.

DA COSTA, Daniel Moura Martins. **Ensemble Baseado em Métodos de Kernel para Reconhecimento Biométrico Multimodal**. 2016. Dissertação (Mestrado em Sistemas de Informação) - Universidade de São Paulo, São Paulo, 2016.

DALAL, Navneet; TRIGGS, Bill. Histograms of Oriented Gradients for Human Detection. In: **International Conference on computer vision & Pattern Recognition (CVPR'05)**. IEEE Computer Society, 2005. p. 886--893.

DE SANTANA, Luciana Maiara Queiroz; ROCHA, Fabio Gomes. Processo de Detecção Facial Utilizando Viola; Jones. **Interfaces Científicas-Exatas e Tecnológicas**, v. 1, n. 1, p. 35-40, 2015.

DO VALE MANUEL, Lucas *et al.* Identificação e Rastreamento de Pessoas por Meio de Imagens Digitais Capturadas a Partir de Câmeras de Vídeo. In: **Colloquium Exactarum**. 2016.

ESTEPA, Adriana Paola Castillo. **Saúde Visual no Trabalho e a Síndrome da Visão do Computador em Professores Universitários**. 2014. Dissertação (Mestrado em Saúde Coletiva) - Faculdade de Ciências Médicas, Universidade Estadual de Campinas, Campinas, 2014.

FERRASSINI, Rafael Barichello. **Sistema de Detecção de Sonolência, por meio de Visão Computacional**. 2014. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica com ênfase em Eletrônica) - Universidade de São Paulo, São Carlos, 2014.

FERREIRA, Alessandro dos Santos. **Redes Neurais Convolucionais Profundas na Detecção de Plantas Daninhas em Lavoura de Soja**. 2017. Dissertação (Mestrado em Ciências da Computação) - Universidade Federal de Mato Grosso do Sul, Campo Grande, 2017.

GONÇALVES, Andreia do Carmo. **Alterações das Propriedades Óticas do Olho com a Utilização de Computadores**. 2018. Dissertação (Mestrado em Optometria Avançada) - Universidade do Minho, [S. I.], 2018.

GONZALEZ, Rafael C. WOODS, Richard E. **Processamento Digital de Imagens**. 3. ed. São Paulo: Pearson Universidades, 2009.

JÚNIOR, Luiz Otávio de Oliveira Souza. **Um Estudo Sistemático Sobre Detecção de Impostor Facial**. 2016. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal da Bahia, Universidade Estadual de Feira de Santana, Salvador - BA, 2016.

LEÃES NETO, Antônio do Nascimento. **Classificação com Algoritmo Adaboost.m1: O Mito do Limiar de Erro de Treinamento**. 2017. Dissertação (Mestrado em Ciência da Computação) - Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, 2017.

LOPES, Eduardo Costa. **Detecção de Faces e Características Faciais**. 2003. Dissertação (Mestrado em Ciência da Computação) - Faculdade de Informática, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre - RS, 2003.

MEDEIROS, Raianny Alves Costa. **Diagnóstico de Enfermagem Risco de Olho Seco e Ressecamento Ocular em Pacientes Adultos em Pós-operatório Imediato**. 2017. Dissertação (Mestrado em Enfermagem) - Universidade Federal do Rio Grande do Norte, Natal, 2017.

MENDES, Charles Corrêa. **Classificação de Imagens de Retina para Auxílio a Diagnóstico de Retinopatia Diabética via Detecção de Exsudatos Duros, com Base em Classificadores Robustos, Máquinas de Vetores de Suporte e Algoritmos com Treinamento de Dicionários**. 2017. Dissertação (Mestrado em Engenharia Biomédica) - Universidade de Brasília, Brasília - DF, 2017.

NEVES, Luiz Ricardo Cruz; D'ARAUJO FILHO, José Jesu Sisnando. Estudo dos Sintomas Visuais na Síndrome Relacionada ao Computador e Efeitos dos Colírios

Lubrificantes em Funcionários do Hospital Universitário Bettina Ferro de Sousa. **Revista Eletrônica Acervo Saúde**, v. 11, n. 8, p. e591-e591, 2019.

OKADA, Hugo Kenji Rodrigues. **Detecção Automática de Fases Temporais de Emoção em Vídeos a partir de Características da Face**. 2018. Dissertação (Mestrado em Informática) - Universidade Federal do Amazonas, Manaus - AM, 2018.

PAJANKAR, Ashwin. **Raspberry Pi computer vision programming**. Packt Publishing Ltd, 2015.

PATEL, S. *et al.* Effect of Visual Display Unit Use on Blink Rate and Tear Stability. **Optom Vis Sci**, v. 68, n. 11, p. 888-892, 1991.

PLATT, John. **Sequential minimal optimization: A fast algorithm for training support vector machines**. 1998.

PICCOLI, Bruno. A Critical Appraisal of Current Knowledge and Future Directions of Ergophthalmology: Consensus Document of the ICOH Committee on 'Work and Vision'. **Ergonomics**, v. 46, n. 4, p. 348-406, 2003.

SÁ, Eduardo Costa. **Síndrome da Visão do Computador e Função Visual em Trabalhadores Usuários de Computador de um Hospital Público Universitário de São Paulo: Prevalência e Fatores Associados**. 2016. Tese (Doutorado em Saúde Ambiental) - Faculdade de Saúde Pública, Universidade de São Paulo, São Paulo, 2016.

SAGONAS, Christos *et al.* 300 Faces In-the-wild Challenge: Database and Results. **Image and Vision Computing**, v. 47, p. 3-18, 2016.

SAMPAIO, Gustavo Scalabrini *et al.* **Desenvolvimento de uma interface computacional natural para pessoas com deficiência motora baseada em visão computacional**. 2018.

SANTIAGO, Hemir Da Cunha. **Reconhecimento de Expressões Faciais Utilizando Estimacão de Movimento**. 2017. Tese (Doutorado em Ciência da Computação) - Universidade Federal de Pernambuco, Recife, 2017.

SARDINHA, Érika A. *et al.* **Um Componente para Detecção de Piscada de Olhos Utilizando a Interação Homem Computador**. 2016.

SHAPIRO, Linda; STOCKMAN, George. **Computer Vision**. 2000.

SILVA, João Paulo *et al.* Lógica Fuzzy Aplicada ao Controle de Intensidade da Luz. **Revista Científica da UNIFENAS**, v. 1, n. 1, 2019.

SZELISKI, Richard. **Computer Vision: Algorithms and Applications**. 2010.

TONG, Yan *et al.* Semi-supervised facial landmark annotation. **Computer Vision and Image Understanding**, v. 116, n. 8, p. 922-935, 2012.

WAGNER, Felipe Rocha. **Análise Antropométrica Semiautomática em Imersão para Pesquisa e Diagnóstico Clínico de Síndromes Dismórficas**. 2017. Dissertação (Mestrado em Computação Aplicada) - Universidade do Vale do Rio dos Sinos, São Leopoldo, 2017.

YAN, Zheng *et al.* Computer Vision Syndrome: A widely spreading but largely unknown epidemic among computer users. **Computers in Human Behavior**, v. 24, n. 5, p. 2026-2042, 2008.

YANG, Ming-Hsuan; KRIEGMAN, David J.; AHUJA, Narendra. Detecting faces in images: A survey. **IEEE Transactions on pattern analysis and machine intelligence**, v. 24, n. 1, p. 34-58, 2002.

ANEXO A – TERMO DE AUTORIZAÇÃO DE IMAGEM.

TERMO DE AUTORIZAÇÃO PARA USO DE IMAGENS (TCFV) (FOTOS E VÍDEOS)

Eu, _____, **AUTORIZO** o Kaique Rijkaard de Sousa Oliveira, coordenador(a) da pesquisa intitulada: sistema para detecção do pestanejar utilizando visão computacional para prevenção da síndrome visual do computador a fixar, armazenar e exibir a minha imagem por meio de foto com o fim específico de inseri-la nas informações que serão geradas na pesquisa, aqui citada, e em outras publicações dela decorrentes, quais sejam: revistas científicas, jornais, congressos, entre outros eventos dessa natureza.

A presente autorização abrange, exclusivamente, o uso de minha imagem para os fins aqui estabelecidos e deverá sempre preservar o meu anonimato. Qualquer outra forma de utilização e/ou reprodução deverá ser por mim autorizada, em observância ao Art. 5º, X e XXVIII, alínea “a” da Constituição Federal de 1988.

O pesquisador responsável Kaique Rijkaard de Sousa Oliveira, assegurou-me que os dados serão armazenados em seu computador pessoal, sob sua responsabilidade, por 5 anos, e após esse período, serão destruídas.

Assegurou-me, também, que serei livre para interromper minha participação na pesquisa a qualquer momento e/ou solicitar a posse de minhas imagens.

Ademais, tais compromissos estão em conformidade com as diretrizes previstas na Resolução Nº. 466/12 do Conselho Nacional de Saúde do Ministério da Saúde/Comissão Nacional de Ética em Pesquisa, que dispõe sobre Ética em Pesquisa que envolve Seres Humanos.

Patos, ____ de ____ de _____.

Assinatura do participante da pesquisa

Assinatura do pesquisador responsável