

SISTEMA EMBARCADO PARA DETECÇÃO E ALERTA DE ATTITUDES DE DESATENÇÃO AO DIRIGIR

Mariana Ruivo Rabello de Lima – ruivomari@gmail.com

Antonio Newton Licciardi Junior (Orientador) – newton.licciardijr@gmail.com

RESUMO

A falta de atenção ao volante é uma das principais causas de acidentes de trânsito em todo o mundo. Este fato impulsionou o estudo e desenvolvimento de tecnologias capazes de detectar a sonolência do motorista. Desta forma, o presente trabalho visa a criação de um dispositivo embarcado capaz de alertar o motorista em estado de cansaço e/ou sonolência através de alarmes sonoros e visuais. Para isso, é feito o monitoramento dos pontos de interesse utilizando técnicas de Visão Computacional, através da biblioteca OpenCV, utilizando algoritmos como *landmarks* e os parâmetros *Eye Aspect Ratio* e *Mouth Aspect Ratio*. São apresentados também os materiais e métodos para desenvolvimento do programa e do protótipo. Através dos resultados obtidos dos testes em campo, fez-se uma Matriz de Confusão capaz de medir o grau de eficiência do dispositivo desenvolvido, que apresentou uma boa eficiência para detecção dos sinais de desatenção ao volante.

Palavras-chave: Visão Computacional. OpenCV. Detecção de sonolência. *Landmarks*.

ONBOARD SYSTEM FOR DETECTION AND ALERT OF INATTENTION WHILE DRIVING

ABSTRACT

Lack of attention behind the wheel is one of the main causes of traffic accidents around the world. This fact prompted the study and development of technologies capable of detecting driver drowsiness. Thus, the present work aims to create an embedded device capable of alerting the driver in a state of tiredness and/or drowsiness through audible and visual alarms. For this, points of interest are monitored using Computer Vision techniques, through the OpenCV library, using algorithms such as Landmarks and Eye Aspect Ratio and Mouth Aspect Ratio parameters. The materials and methods for developing the program and the prototype are also presented. Through the results obtained from the field tests, a confusion matrix was made capable of measuring the degree of efficiency of the device developed, which showed good efficiency for detecting signs of inattention while driving.

Key-words: Computer Vision. OpenCV. Drowsiness detection. Landmarks.

1 INTRODUÇÃO

Segundo a World Health Organization (WHO, 2018, p. 5), mais de 1,35 milhão de vidas são perdidas a cada ano no mundo devido aos acidentes de trânsito. Entre as principais causas estão a falta de atenção, ingestão de álcool, desobediência à sinalização, excesso de velocidade e defeitos no veículo. O Brasil ocupa a quarta posição de trânsito mais violento do continente americano (WHO, 2018). O país registra também um elevado número de acidentes por dia, provocados em quase 90% por falhas humanas (DNIT, 2021).

Os fabricantes têm investido cada vez mais no desenvolvimento de inovações que visam maior proteção contra acidentes. Exemplo disso são tecnologias já existentes em alguns carros da Volkswagen, como: controle adaptativo de velocidade e distância, assistente de partida em subida/aclive, assistente de estacionamento, indicador de perda de pressão dos pneus etc (VW, 2021).

Para Balbinot; Zaro; Timm *apud* Pirito (2011, p.27): o ato de dirigir requer “(...) memória, atenção, avaliação do ambiente e outras habilidades visuoespaciais, verbais e de processamento de informações, tomada de decisões e resolução de problemas.”. De acordo com o National Highway Traffic Safety Administration (NHTSA, 2021), departamento de transportes dos Estados Unidos, em média 100 mil acidentes por ano são provocados devido à fadiga e sonolência ao dirigir. Estes números não são precisos, pois não há como checar se o motorista, de fato, dormiu ao volante.

Segundo Queiroz (2011, p.2), o desenvolvimento de sistemas que monitoram o comportamento dos motoristas através da visão computacional é uma das soluções de segurança no trânsito de técnica não intrusiva, auxiliando na redução do número de acidentes. Este tema é alvo de pesquisas e trabalhos que visam resolver o problema de distração e sonolência dos motoristas. Entretanto, ainda há grande dificuldade em encontrar dispositivos no mercado que cumpram o papel de detecção de sonolência e distração dos motoristas com boa confiabilidade, eficiência e baixo custo.

O presente trabalho pretende desenvolver uma ferramenta por meio de reconhecimento facial, baseado no monitoramento dos movimentos dos olhos e boca, através de linguagens de programação, a fim de alertar o motorista em estado de desatenção ao volante. Tal aspecto é propício para a redução do número de acidentes de trânsito.

Como objetivos específicos deste trabalho, ressaltam-se:

- a) encontrar e estudar as características de bases de dados abertas com imagens pessoas em estado de atenção e desatenção;
- b) detalhar as ferramentas adequadas para o processamento de imagens;
- c) estudar as plataformas viáveis para embarcar o dispositivo dentro de um automóvel;

- d) definir as linguagens e ferramentas adequadas para o reconhecimento facial, que caracterize desatenção do motorista;
- e) identificar os sistemas de alarme comumente usados para soluções a fim de recapturar a atenção do motorista;
- f) efetuar a prototipação e testes do sistema em campo e/ou laboratório;
- g) caracterizar a eficiência dos algoritmos através da matriz ROC (*do inglês: Rank Order Clustering*). Eventualmente, efetuar os ajustes necessários e possíveis.

2 METODOLOGIA

A metodologia do trabalho está dividida em etapas que atendem os objetivos detalhados na Seção 1. A seguir são apresentadas as etapas do projeto.

Etapa 1 – Levantamento de imagens

Informações sobre conjunto de imagens de pessoas em estado de atenção e desatenção são levantadas, preferencialmente, a partir de bases de dados abertas existentes na internet de trabalhos anteriores, como as mencionadas na Subseção 3.1. Os testes são realizados respeitando José Silva (2015), que destaca que para câmeras de resolução média (640 x 480, 1024 x 768), o importante é obter imagens de pessoas com características faciais distintas, dependendo do algoritmo a ser adotado. Alternativamente, poderão ser obtidas imagens de voluntários, assim com a autorização devida e termo assinado de uso de imagens devido, em respeito à Constituição Federal, artigo 5, inciso X.

Etapa 2 – Linguagens Computacionais, Bibliotecas, Visão Computacional, Eficiência

Nesta etapa, são levantadas e estudadas as linguagens e os algoritmos/bibliotecas de visão computacional, a partir de recentes materiais científicos disponíveis, como os mencionados nas Subseções a seguir. A partir dos recomendados, levantar os que possuem melhor eficiência. Por algoritmos eficientes, entenda que são aqueles que maximizam os critérios definidos na Subseção 3.8, sobre a Matriz ROC. Linguagens, algoritmos e bibliotecas são usados, baseados nos resultados obtidos de trabalhos prévios e/ou ensaios computacionais. Cabe mencionar que ajustes são trabalhados no programa desenvolvido, a fim de refinar resultados e eficiência, aspectos discutidos na Seção 4.

Etapa 3 – Estudo de Câmeras, Atuadores e Microcontroladores

A partir de trabalhos científicos de relevância, como os apresentados na Seção 3, decide-se por utilizar câmera de média resolução, acoplada ao microcontrolador, para a captura e reconhecimento de imagens de motoristas em tempo real. São levantados também os atuadores acoplados a sistemas embarcados recomendados para alertas aos motoristas em atitudes ou estados

de risco na direção. Escolhe-se o melhor conjunto de atuadores baseados nos dispositivos de mercado.

Face aos pontos de atenção mencionados, pode-se obter uma plataforma com boa capacidade de processamento de imagens em tempo real para aplicação no sistema embarcado. Por boa, compreenda como as plataformas de computação capazes de suportar adequadamente o processamento definido na Etapa 2. Para os levantamentos de custo, são consultadas informações junto a importadores e/ou fabricantes no Brasil, apresentados na Subseção 4.3.

Etapa 4 – Prototipação e testes

Nesta etapa, implementa-se o protótipo do sistema embarcado e ensaios preparatórios de laboratório e de campo são implementados para ajustes de hardware e software, detalhados e discutidos na Seção 4.

3 REVISÃO DA LITERATURA

3.1 BASE DE DADOS

A etapa da escolha da base de dados preliminar, apresentada a seguir, leva em consideração alguns fatores que serão estudados nas fases do TCC: qualidade dos dados, posicionamento das imagens, iluminação e quantidade de imagens. Além disso, elas devem ser capazes de ajudar a reconhecer o estado de sonolência e distração do motorista ao volante. As bases de dados pesquisadas como materiais de revisão são: *Closed Eyes In The Wild (CEW)*, *Yale Face Database B*, *Georgia Tech* e *Sheffield*.

3.2 VISÃO COMPUTACIONAL

Visão computacional é uma área de atuação do campo da inteligência artificial que tem como objetivo recriar o mecanismo da visão humana por meio de softwares e hardwares. Segundo Backes e Junior (2019, p.1), “podemos definir visão computacional como a área de estudo que tenta repassar para máquinas a incrível capacidade da visão.”.

Para Carvalho Junior (2018, p.25) o maior desafio do processamento de imagens é conseguir que as máquinas façam um trabalho tão bom quanto o ser humano, quando se trata da identificação de padrões.

O sistema de visão computacional é constituído das seguintes fases: aquisição, processamento de imagens, segmentação, extração de características e reconhecimento de padrões.

3.2.1 Aquisição

Constitui a primeira etapa de um sistema de Processamento Digital de Imagens. Segundo Backes e Junior (2019, p.2), trata-se do processo de captação de imagens, por meio de periféricos como filmadoras, máquinas fotográficas e webcam. Essas imagens podem ser bidimensionais, tridimensionais ou uma sequência de imagens. (NORONHA, 2019).

3.2.2 Processamento de imagens

As imagens capturadas necessitam ser filtradas e convertidas para algum tamanho e/ou formato, uma vez que do processo de aquisição surgem ruídos e imperfeições. O processamento de imagens consiste em melhorar a imagem, corrigindo imperfeições, removendo ruídos, suavizando a imagem e salientando bordas (AZEVEDO; CORREA, 2019, p.30). Segundo Marengoni e Stringhini (2010, p.127),

Os ruídos podem aparecer de diversas fontes, como por exemplo, o tipo de sensor utilizado, a iluminação do ambiente, as condições climáticas no momento da aquisição da imagem, a posição relativa entre o objeto de interesse e a câmera. Note que ruído não é apenas interferência no sinal de captura da imagem, mas também interferências que possam atrapalhar a interpretação ou o reconhecimento de objetos na imagem.

3.2.3 Segmentação

A segmentação de imagens consiste em subdividir uma imagem em um conjunto de regiões que a compõem. O nível de detalhes neste processo varia conforme a quantidade de precisão que se deseja obter sobre a imagem, sendo feito com base em determinadas características do objeto de interesse como, por exemplo, cor, proximidade e formas. As principais técnicas de segmentação são: detecção por borda, detecção por corte e por crescimento de região.

3.2.4 Extração de características

Um conjunto de características é obtido por meio do objeto de interesse. A imagem é transformada em um conjunto de dados, a partir de uma codificação numérica, denominado vetor de características.

3.2.5 Reconhecimento de padrões

Este processo identifica a posição e orientação, reconhecendo os padrões do objeto de interesse. Segundo Marengoni e Stringhini (2010, p.144), para realizar o reconhecimento de imagens, é necessária uma base de conhecimento dos objetos a serem reconhecidos, que podem ser identificados através do aprendizado de máquina. O objetivo da aprendizagem de máquina está na programação de computadores para identificar padrões ou comportamentos, partindo de dados que servem como exemplos, ou seja, por meio de um treinamento inicial (SOUZA, 2015, p.24).

3.3 SISTEMAS EMBARCADOS

Sistemas eletrônicos embarcados são dispositivos computacionais que funcionam como computadores, possuindo memória, processador, elementos de entrada e saída, mas que executam tarefas específicas pré-definidas nos dispositivos (MARWEDEL, 2021). Um sistema embarcado é uma combinação de hardware e software e permite ter sua funcionalidade alterada através da

reconfiguração via software. Dentre os principais sistemas embarcados utilizados na Engenharia, podemos citar: Raspberry Pi, Arduino e FPGA.

Segundo Paranhos (2019), Noronha (2019), Glasenapp e Hoppe (2019), o uso do Raspberry Pi é factível para trabalhos de reconhecimento de imagens. Assim, visando o desenvolvimento de um protótipo para detecção de sonolência, o Raspberry Pi é apresentado como uma boa opção, devido à sua capacidade de processamento e baixo custo, além de ser um dispositivo prático e acessível.

O Raspberry Pi 3B+ é um microcontrolador que possui processador BCM2837BO da Broadcom Quad-Core Cortex-A53 de 1,4 GHz, unidade de processamento gráfico (GPU), memória RAM de 1GB, quatro portas USB 2.0, saída HDMI, conector GPIO de 40 pinos, entrada para cartão microSD, saídas SCI e DSI para câmera e display, porta microUSB para alimentação (5V / 2,5A), bluetooth 4.2 e conector para periféricos de (HDMI, USB, microUSB, microSD, áudio, ethernet) e módulo Wi-Fi (RICHARDSON; WALLACE, 2013, p.24).

O Raspberry Pi permite também trabalhar com inúmeras bibliotecas voltadas para o reconhecimento facial, como OpenCV e Dlib. Caso o Raspberry Pi não seja capaz de efetuar em tempo real a detecção de sonolência e/ou distração, o dispositivo embarcado alternativo será o FPGA (MATAI; IRTURK; KASTNER, 2011).

O FPGA (do inglês: *Field Programmable Gate Array*) é um dispositivo lógico programável que permite a conexão de circuitos digitais que realizam funções comuns com elevado grau de flexibilidade e velocidade de processamento (UFOP, 2021).

3.4 LINGUAGENS DE PROGRAMAÇÃO

As linguagens de programação surgiram com o objetivo de implementar instruções matemáticas de forma que o computador fosse capaz de compreender e realizar estas operações. Face à proposta temática, três linguagens de programação comumente utilizadas no desenvolvimento de sistemas embarcados em engenharia são (MARWEDEL, 2021): Python, C e VHDL.

Para o reconhecimento de imagens em sistemas embarcados, o microcontrolador com maior capacidade de processamento é o Raspberry Pi (GUNAWAN *et al.*, 2017). No Raspberry Pi, os algoritmos de reconhecimento de imagens estão disponíveis em bibliotecas relacionadas a linguagem Python (PARANHOS, 2019). Este foi o critério de decisão que conduziu a escolha da linguagem de programação. Destacam-se para o corrente trabalho a possibilidade de uso das seguintes bibliotecas:

- a) OpenCV (*Open Source Computer Vision*): biblioteca de software de visão computacional e aprendizado de máquina em código aberto;

- b) Dlib: biblioteca multiplataforma de código aberto desenvolvida na linguagem C++ e possui diversos algoritmos e ferramentas de aprendizado de máquina e processamento digital de imagens (PARANHOS, 2019, p. 46);
- c) NumPy: projeto de código aberto que visa habilitar a computação numérica com Python, permitindo trabalhar com arranjos, vetores e matrizes de N dimensões, de sintaxe semelhante ao Matlab, mas com maior eficiência (PYSCIENCE, 2019);
- d) Imutils: funções de conveniência para simplificar as funções básicas de processamento de imagens;
- e) SciPy: biblioteca *open source* de computação científica que contém ferramentas para estatística, otimização, interpolação, processamento de imagens e sinais (SCIPY, 2019).

Nas Subseções 3.5 a 3.7 a seguir, são apresentados os algoritmos típicos de reconhecimento de imagem, bem como a medição de eficiência do mesmo.

3.5 ALGORITMO DE VIOLA JONES

Desenvolvido pelos pesquisadores Paul Viola e Michael Jones, em 2001, o algoritmo é uma técnica de reconhecimento de padrões que consiste em encontrar partículas de imagem que apresentem diferença de contraste. O algoritmo combina basicamente quatro conceitos-chave:

- a) Características *Haar-like* retangulares: realizam a codificação dos contrastes existentes entre as regiões orientadas da imagem, a fim de extrair as informações necessárias para interpretação do rosto humano ou outra característica (DINIZ *et al*, 2012, p.4);
- b) Imagem Integral: o algoritmo tem como objetivo o mapeamento de texturas e é capaz de calcular a soma dos valores em um subconjunto retangular de uma matriz de forma rápida;
- c) Algoritmo de aprendizagem (*AdaBoost*): é método de aprendizado de máquina que combina de forma ponderada vários classificadores fracos com taxa de acerto acima de 50% para obter um classificador forte (FREUND; SCHAPIRE, 1997, p.130);
- d) Um classificador em cascata: funciona como uma concatenação, em uma sequência definida em várias etapas. Cada estágio da cascata possui um classificador forte, que por sua vez contém um conjunto de classificadores fracos (PARANHOS, 2019, p. 36). Os classificadores são denominados fracos por não haver garantia de que sua classificação seja ótima (RODRIGUES, 2012, p.25). No treinamento é empregado o algoritmo *AdaBoost*, onde é verificado se uma dada janela contém o resultado obtido ou não (Figura 1). Assim, uma janela é descartada imediatamente ao falhar em qualquer um dos estágios. Seu principal objetivo é um classificador simples que possui alta taxa de

detecção no início da cascata que permite rejeitar as janelas negativas cedo e poupar processamento (SILVA, Julio, 2018, p.29).

Figura 1 – Classificador em Cascata



Fonte: SILVA, Julio (2018).

Com a face bem definida, é interessante obter as regiões de interesse para detecção de sonolência com precisão.

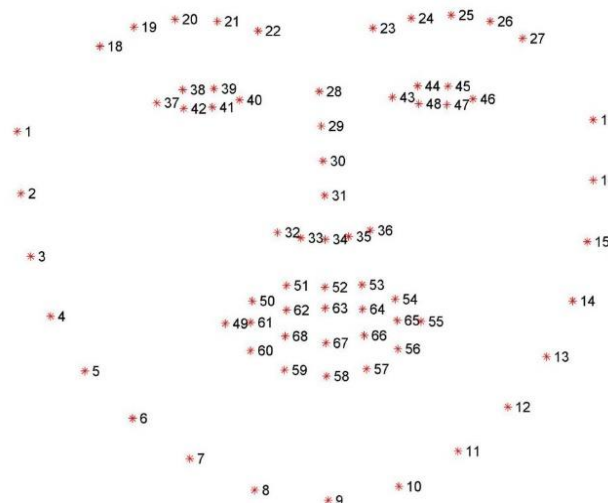
3.6 DETECÇÃO DE PONTOS DE REFERÊNCIA FACIAIS

A detecção dos pontos de referência faciais, também conhecidos como *Facial Landmarks*, é uma técnica que busca identificar os principais pontos de região da face (olhos, boca, nariz, sobrancelha) e suas expressões faciais com um alto índice de precisão (PARANHOS, 2019, p. 36). A biblioteca Dlib, vista na Subseção 3.4, é capaz de encontrar 68 coordenadas (x,y) que mapeiam as estruturas faciais da face de uma pessoa. Obtidas através do treinamento de um modelo de predição chamado iBUG300-W, os dados são armazenados em vetor que podem ser utilizados para cada área que se deseja obter, escolhendo o intervalo necessário (Figura 2).

Os vetores interessantes para detecção de sonolência possuem os seguintes pontos de referência:

- a) boca [48,68]
- b) olho direito [36, 41]
- c) olho esquerdo [42, 47]

Figura 2 – *Facial Landmarks*

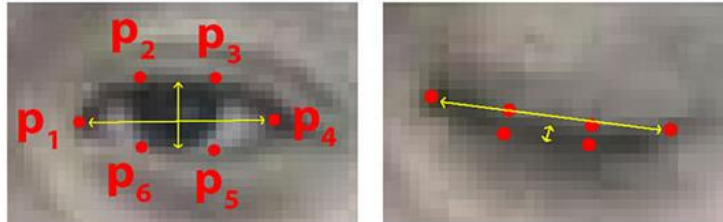


Fonte: PYIMAGESEARCH (2017).

3.7 EYE ASPECT RATIO (EAR) E MOUTH ASPECT RATIO (MAR)

Como foi visto na Subseção 3.6, é possível usar os *landmarks* para mapear importantes regiões da face com precisão. No contexto deste trabalho, é de grande interesse o uso das regiões que correspondem aos olhos esquerdo e direito, além da boca. A medida de *Eye Aspect Ratio* (EAR) é calculada através dos pontos de altura e da largura do olho (Figura 3).

Figura 3 – Pontos de referência dos olhos



Fonte: PYIMAGESEARCH (2017).

Para calcular o EAR, utiliza-se a Equação 1:

$$EAR = \frac{|p_2 - p_6| + |p_3 - p_5|}{2 * |p_1 - p_4|} \quad (1)$$

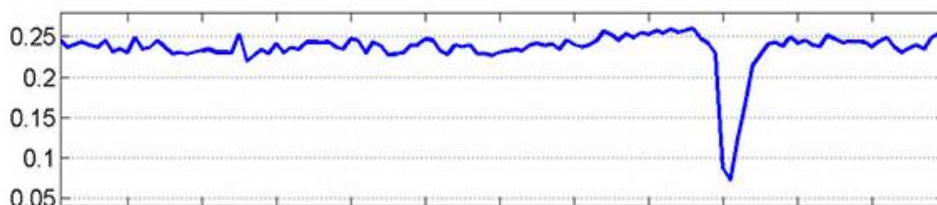
onde:

p_1 a p_6 = coordenadas ilustradas em um plano 2D, representadas pelos pontos [36,41] e [42,47] do vetor *landmarks*.

O numerador da equação calcula a distância entre os pontos de referência verticais dos olhos, enquanto o denominador pondera o cálculo da distância entre os pontos horizontais dos olhos, uma vez que há dois conjuntos de pontos verticais para um conjunto de pontos horizontais (PARANHOS, 2019, p.38)

Sendo assim, o EAR é constante quando um olho permanece em um estado pré-estabelecido via software e se altera a partir do momento em que o olho é fechado. Em média, a medida para o olho aberto varia de 0,25 a 0,32. Logo, para um valor abaixo do estabelecido, é considerado que o olho se encontra fechado, como sugere a Figura 4.

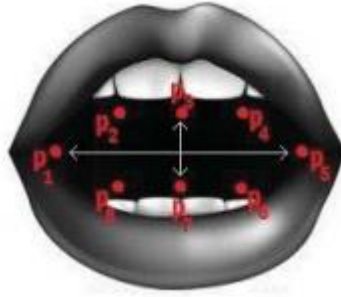
Figura 4 – Gráfico do EAR



Fonte: PYIMAGESEARCH (2017).

De forma análoga ao EAR, o MAR utiliza os pontos de referência [48,68] para calcular a relação de abertura da boca. Esses dados são de grande relevância para detectar o grau de cansaço de um indivíduo, por exemplo. As coordenadas podem ser vistas na Figura 5.

Figura 5 – Pontos de referência da boca



Fonte: PARANHOS (2019).

O MAR é calculado através do cálculo de subtração da mediana do lábio superior em relação à mediana do lábio inferior, determinando se a boca está aberta ou fechada para determinado valor, conforme Equação 2.

$$\text{MAR} = \text{mean}(\text{top}_{\text{lip}}) - \text{mean}(\text{low}_{\text{lip}}) \quad (2)$$

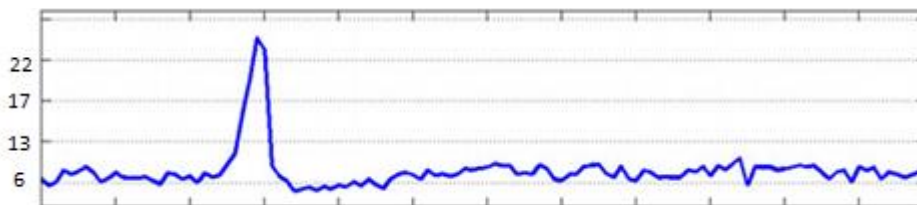
sendo:

top_{lip} = medida da parte superior da boca;

low_{lip} = parte inferior da boca.

Desta forma, o MAR é constante quando a boca permanece fechada e, de acordo com o parâmetro estabelecido através da média do grau de abertura da boca, o valor é alterado. Em média, uma boca aberta atinge um valor de MAR acima de 22, considerando um bocejo, como é apresentado na Figura 6.

Figura 6 – Gráfico do MAR



Fonte: adaptado de PYIMAGESEARCH (2017).

3.8 MATRIZ ROC

O algoritmo ROC pode ser traduzido como “algoritmo por ordem de grandeza” e é utilizado para organizar e avaliar classificadores, por meio de gráficos que permitem visualizar o limiar entre taxas de acertos e erros dos algoritmos. Com base em um conjunto de amostras, uma instância pode admitir quatro situações: verdadeiro positivo, verdadeiro negativo, falso positivo e falso negativo. Para este conjunto de instâncias, pode-se criar uma Matriz de Confusão (MC) de duas classes, como é apresentado na Figura 7.

Figura 7 - Matriz de Confusão de um conjunto de instâncias

		Classe verdadeira	
		Positivo	Falso
Classe esperada	Positivo	Verdadeiro Positivo	Verdadeiro Negativo
	Falso	Falso Positivo	Falso Negativo

Fonte: autor.

Essas instâncias são importantes para avaliar a eficiência de algoritmos e dispositivos embarcados: os índices de acerto (verdadeiro positivo e verdadeiro negativo) e de falhas (falso negativo e falso positivo). A intenção básica é sempre maximizar os acertos e minimizar as falhas.

A partir da Matriz de Confusão é possível construir métricas importantes como:

- a) precisão: percentual de preditos positivos corretos dentro de todos casos preditos positivos, demonstrada na Equação 3;
- b) acurácia: proporção de acertos do modelo, representada na Equação 4.

$$\text{Precisão} = \frac{VP}{VP+FP} \quad (3)$$

$$\text{Acurácia} = \frac{VP}{VP+VN+FP+FN} \quad (4)$$

em que:

VP = verdadeiro positivo;

VN = verdadeiro negativo;

FP = falso positivo;

FN = falso negativo.

2.9 SISTEMAS DE ALARME

Alguns modelos de sensores para alertar motoristas em caso de desatenção ao dirigir já são comercializados. O sensor de fadiga DCM-01 da Tecnologia GPS, comercializado no mercado, possui funções como: alarme de distração, alarme de velocidade, alarme de fadiga e infravermelho para funcionamento à noite (Tecnologia GPS, 2021). A proposta da empresa foi disponibilizada no valor de R\$2.710,10 para o dispositivo e R\$280,00 para a almofada com alarme vibratório opcional (pesquisa realizada em maio de 2021). Outros sensores de fadiga encontrados foram os das empresas Marpress (MARPRESS, 2021) e Informeservice (INFORMESERVICE, 2021). Os atuadores típicos utilizados são alertas visuais e/ou sonoros.

Das soluções encontradas no mercado, cabe realçar os seguintes pontos comuns:

- c) são fornecidas características técnicas básicas sobre a função do dispositivo, sem mencionar qualquer aspecto relacionado à eficiência do produto;
- d) para a que teve o custo do produto divulgado, pôde-se observar que, somando-se ao custo de instalação, ele não é de baixo custo;
- e) nenhum comentário está disponível sobre a confiabilidade das soluções.

Face aos aspectos mencionados, está clara a relevância técnico-científica de estudar e prototipar um dispositivo que seja capaz de identificar e alertar situações de distração ao volante, confiável, de eficiência mensurada e de baixo custo para o mercado brasileiro. Atrela-se a este aspecto a relevância social na obtenção de sistema embarcado como o descrito.

4 RESULTADOS E DISCUSSÃO

Esta Seção detalha os resultados obtidos após a execução dos códigos e faz uma análise comparativa entre as diferentes técnicas e parâmetros utilizados. Durante o desenvolvimento do sistema proposto, foram realizados inúmeros ensaios e, após ser finalizado, foram efetuadas provas com diversas pessoas a fim de avaliar o desempenho do mesmo. Com o objetivo de determinar a adequação das funções disponíveis para reconhecimento facial, disponíveis no OpenCV, alguns ensaios preparatórios foram realizados e encontram-se explicados na Subseção 4.1 a seguir.

4.1 ENSAIOS PREPARATÓRIOS

Inicialmente, foram feitos testes de detecção da face e dos olhos utilizando o classificador *Haar Cascade* do OpenCV, que utiliza como base o algoritmo de Viola Jones, visto na Subseção 3.5. Os testes foram feitos utilizando imagens do banco de dados da *Georgia Tech*, que contém imagens de 50 pessoas em poses frontais e inclinadas com diferentes expressões faciais, escalas e iluminação (GT, 2021) (Figura 8a).

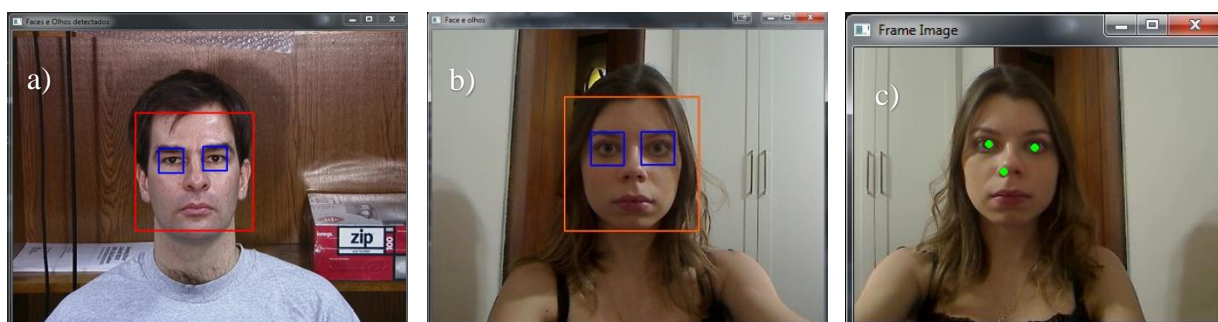
O objetivo destes testes era treinar a linguagem Python de programação junto a técnicas de reconhecimento facial, utilizando o *PyCharm* – uma IDE (ambiente de desenvolvimento integrado) Python. Posteriormente, os testes de detecção foram feitos utilizando imagens capturadas via *webcam*, a fim de trabalhar com o reconhecimento em tempo real para que pudesse ser usado dentro do veículo (Figura 8b).

Para realização da detecção da sonolência, pensou-se em utilizar a transformada *Hough Circle Transform* do OpenCV. Esta transformada é capaz de detectar círculos por meio do seu raio e suas coordenadas de centro (X_c , Y_c). Para armazenar estes três parâmetros, o OpenCV utiliza o método *Hough Gradient*, que utiliza as informações de gradiente nas bordas e localização de possíveis centros, e encontra o melhor raio para cada centro (OPENCV, 2021). Assim, a função seria capaz de detectar o estado da íris por meio do círculo detectado. Entretanto, nos testes, a

função não foi capaz de identificar os olhos com precisão, devido à necessidade de ajuste dos parâmetros que divergiam com cada imagem inicialmente testada. Além disso, verificou-se que a transformada conseguia detectar melhor imagem de objetos circulares do que imagem de voluntários para detectar os olhos na imagem.

Em outro teste, utilizou-se operações morfológicas do OpenCV, que são operações simples baseadas na forma da imagem e geralmente executadas em imagens binárias (OPENCV2, 2016). Para o uso dessa operação são necessárias duas entradas: a imagem original e a outra chamada de *kernel*, que decide a natureza da operação (OPENCV2, 2016). Dentre as operações morfológicas existentes, testou-se utilizar a dilatação e a abertura. Essas operações realizam, respectivamente, as seguintes funções: aumentam os pixels do primeiro plano, unindo as partes quebradas de uma imagem e removem ruídos. A partir da obtenção da imagem da íris, foram detectados os contornos e desenhado um círculo ao redor dela. Entretanto, ao treinar a função com a *webcam*, o programa identificou inúmeras vezes as narinas como se fossem olhos, como pode-se observar na Figura 8c.

Figuras 8 a, b, c – Ensaios preparatórios

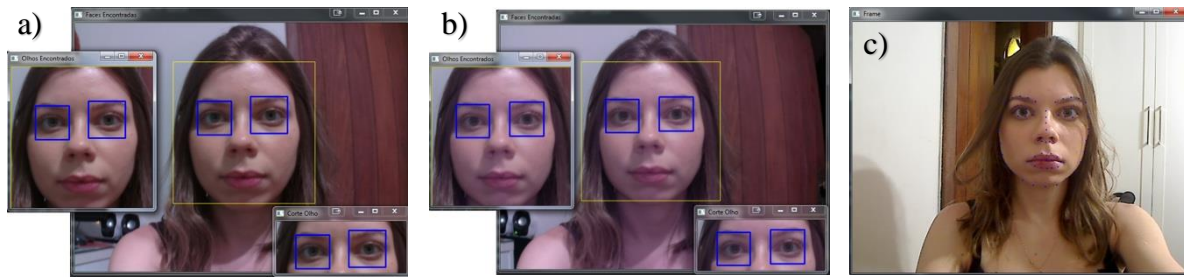


Fonte: autor.

Iniciando os testes com a resolução da câmera do Raspberry Pi, foram feitas capturas de imagens para treinar a detecção da face (Figura 9a). Para estes testes, a face detectada era recortada em outra imagem, assim como os olhos. Entretanto, devido à iluminação diferente em cada dia e horário, nem sempre era possível que o recorte dos olhos fosse feito de maneira eficiente nos testes. Por eficiente entenda que os dois olhos são recortados para análise. Visando diminuir estes erros de detecção, foram implementadas luzes infravermelhas à câmera do Raspberry Pi (Figura 9b).

A técnica de reconhecimento facial do OpenCV é eficiente, mas não é capaz de detectar as expressões faciais, por exemplo. Por este motivo, optou-se por utilizar a técnica de *landmarks*, vista na Subseção 3.6, para identificar as regiões de interesse do rosto em tempo real (Figura 9c).

Figuras 9 a, b, c – Ensaios Preparatórios



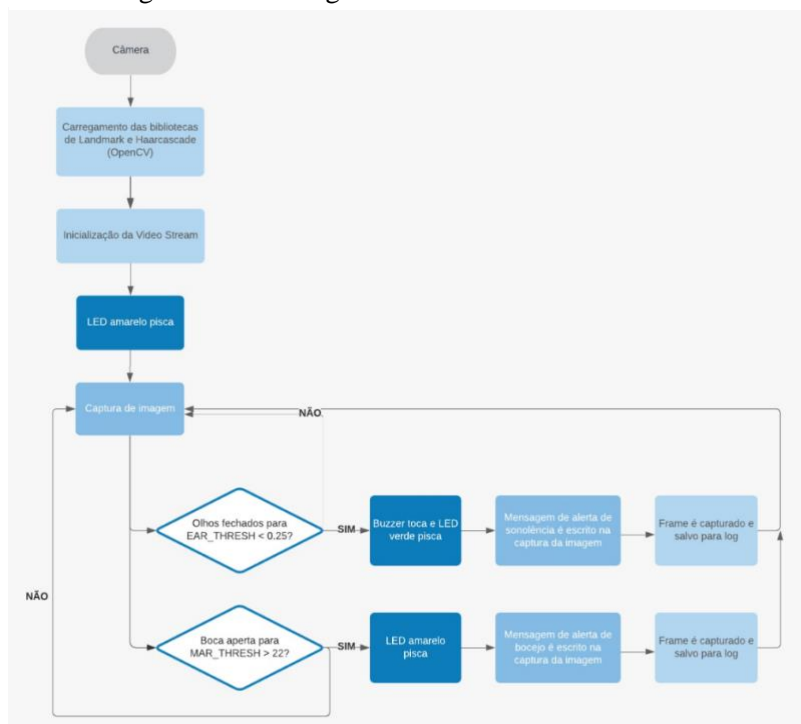
Fonte: autor.

4.2 CÓDIGO DESENVOLVIDO

Para o desenvolvimento do projeto, aperfeiçoou-se o código utilizando a detecção de faces do OpenCV em conjunto aos *landmarks* para captar as regiões de interesse para monitoramento do estado dos olhos e da boca, através dos cálculos de EAR e MAR, vistos na Subseção 3.7. O código do desenvolvido encontra-se disponível na página do GitHub e assinado pelo DOI (10.5281/zenodo.5748515).

A Figura 10 apresenta o fluxograma do sistema desenvolvido.

Figura 10 – Fluxograma do software desenvolvido



Fonte: autor.

As etapas que constituem o sistema de detecção de sonolência e desatenção estão descritas na Tabela 1 a seguir:

Tabela 1 – Etapas do código desenvolvido

ETAPAS	DESCRIÇÃO
Etapa 1	Carregar as bibliotecas essenciais para o funcionamento do programa

Etapa 2	Definir das fórmulas para o cálculo de EAR e MAR;
Etapa 3	Definir o argumento da webcam esperado para o programa;
Etapa 4	Estabelecer os parâmetros de EAR <i>threshold</i> (valor limite para definição de sonolência) e frames consecutivos (valor limite do número de quadros para detecção de sonolência), do MAR <i>threshold</i> (valor limite para definição de cansaço/bocejo) e o contador (número total de frames consecutivos em que o usuário possui a proporção do olho abaixo do EAR THRESH);
Etapa 5	Instanciar as bibliotecas OpenCV para o classificador HaarCascade e Dlib para os <i>landmarks</i> ;
Etapa 6	Inicializar a <i>stream</i> de vídeo da câmera do Raspberry Pi e piscar LED amarelo para indicar inicialização da gravação;
Etapa 7	Inicializar o loop sobre os quadros na stream de vídeo;
Etapa 8	Redimensionar o frame para exibição da monitoração, converter a captura para a escala de cinza e realizar a detecção do rosto utilizando o classificador HaarCascade;
Etapa 9	Realizar outro loop sobre o rosto detectado e determinar os pontos das regiões de interesse utilizando os <i>landmarks</i> , convertendo o resultado em uma matriz NumPy;
Etapa 10	Extrair as coordenadas (x,y) dos olhos e boca, e realizar o contorno das regiões detectadas para melhor identificação no frame;
Etapa 11	Inserir no frame os valores dos parâmetros de EAR e MAR, conforme a leitura de detecção, além de data e hora;
Etapa 12	Verificar se a proporção do EAR calculado está abaixo do EAR THRESH definido e verificar se o contador excede o valor consecutivo de <i>frames</i> estabelecido, indicando que a sonolência foi detectada. Piscar o LED verde para sinalização e soar o alarme sonoro pelo <i>buzzer</i> . No frame, inserir o texto de alerta de sonolência e realizar a captura do frame para coleta de logs;
Etapa 13	Verificar se a proporção do MAR calculado está acima do MAR THRESH definido, indicando que o bocejo foi detectado. Piscar o LED amarelo para sinalização. No frame, inserir o texto de alerta de bocejo e realizar a captura do frame para coleta de logs.

Fonte: autor.

4.3 MATERIAIS UTILIZADOS

O microcontrolador escolhido para o desenvolvimento do projeto foi o Raspberry Pi 3B+, devido à sua capacidade de processamento, tamanho compacto, custo e facilidade de uso.

Para realizar o *stream* de vídeo, optou-se por utilizar o módulo de câmera de visão noturna, composto por duas luzes de LED infravermelho, a fim de aumentar a intensidade luminosa em ambientes escuros. A câmera possui abertura (f): 1.8, distância focal de 3.6 mm ajustável, diagonal de 75.7 graus e resolução do sensor de 1080p. O atuador é conectado ao microcontrolador através do cabo plano flexível na interface serial para câmera (CSI).

Os atuadores escolhidos para compor o sistema de alerta foram: LEDs de sinalização (amarelo para alertar bocejo e verde para alertar sonolência) e *buzzer* (alarma quando a sonolência é detectada, junto ao LED verde).

O protótipo, feito na impressora 3D, foi desenvolvido para acoplar os LEDs, *buzzer* e a câmera para monitoração, fixada ao microcontrolador por meio de um braço mecânico. A Figura 11 ilustra o protótipo desenvolvido.

O protótipo teve seu custo baseado nos valores investidos para o desenvolvimento do mesmo. A placa Raspberry Pi custa em média R\$400,00, o conjunto de atuadores (2 LEDs e *buzzer*) para sinalização em torno de R\$3,00, a câmera de visão noturna junto com o cabo R\$80,00 e o *case* na impressora 3D em média de R\$100,00. Sendo assim, o protótipo tem seu custo avaliado em torno de R\$583,00.

Figura 11 – Protótipo do hardware



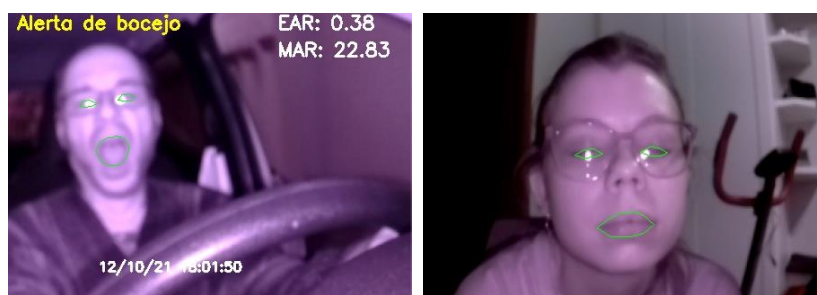
Fonte: autor.

4.4 EFICIÊNCIA DO PROTÓTIPO

Foi solicitado a voluntários que utilizassem o sistema embarcado dentro do veículo, a fim de que se pudesse realizar os ajustes de hardware e software necessários e as medições para cálculo de eficiência do protótipo.

Os testes foram realizados em condições de luminosidade variadas, tanto de dia quanto de noite e, para essas duas condições, o sistema foi capaz de detectar a sonolência de forma semelhante, devido ao uso da luz de IR em conjunto com a câmera. Nos ensaios iniciais, percebeu-se que a luz IR estava com uma alta intensidade e, nos testes utilizando óculos de grau, essa luz era refletida nas lentes, dificultando a detecção dos olhos em estado fechado (Figura 12).

Figura 12 – Reflexão do IR nas lentes de óculos de grau



Fonte: autor.

Para corrigir a intensidade de luz refletida nos óculos e o clarão na pele dos usuários, diminuiu-se a intensidade no potenciômetro da luz IR, permitindo uma identificação da face com maior precisão. (Figura 13).

Figura 13 – Correção da intensidade do IR



Fonte: autor.

Para uma boa detecção de sonolência, a distância ótima do protótipo ao motorista deve ser de 50cm a 70cm, podendo ser colocada tanto na parte lateral do carro (ao lado da janela) quanto na parte central do carro. Por boa entenda que o dispositivo seja capaz de detectar o estado dos olhos e da boca de acordo com o parâmetro de EAR e MAR ajustados (Figura 14).

Figura 14 – Distâncias de 50cm e 70cm do protótipo respectivamente



Fonte: autor.

Os testes foram realizados com quatro voluntários a fim de se obter uma média das características faciais de cada um deles para determinar os parâmetros de EAR e MAR. Em todos os ensaios, foram testadas as condições citadas nesta Subseção: diferentes iluminações (naturais, artificiais e ausência de iluminação), uso de óculos de grau e diferentes posições e distâncias do protótipo (Figura 15 e Figura 16).

Figura 15 – Testes realizados com voluntários



Fonte: autor.

Figura 26 – Testes realizados com voluntários



Fonte: autor.

Para detecção do estado de sonolência e bocejo, foram utilizados os parâmetros de EAR e MAR definidos em 0.25 e 22, respectivamente. Assim, quando os olhos foram detectados com um valor menor que 0.25, o *buzzer* alertou e o LED verde sinalizou a sonolência do motorista. Quando a boca foi aberta, em um valor acima de 22 para MAR, o LED amarelo sinalizou o bocejo do motorista.

Os testes realizados apresentaram algumas variações, entre eles:

- algumas pessoas tendem a colocar a mão à frente da boca enquanto bocejam, o que impossibilita a detecção do movimento da boca;
- devido à características faciais de alguns voluntários, há certo sombreamento na região dos olhos, causado pelo excesso de pálpebras. Este fato conduziu à incorreta identificação do movimento dos olhos e, como alternativa, foi colocada mais luz sobre a face do voluntário para a detecção.
- em condições de pouca iluminação, como os testes realizados na parte da noite, encontrou-se maior dificuldade na detecção do contorno das regiões de interesse. Condição que levou o usuário a se aproximar do dispositivo e/ou incidir maior iluminação no veículo.

Além dos ensaios realizados, foram feitos 320 testes para obter uma Matriz de Confusão (Tabela 2), que admite quatro situações:

- verdadeiro positivo: olhos fechados são detectados como fechados e boca aberta é detectada como aberta;
- verdadeiro negativo: olhos fechados são detectados como abertos e boca aberta é detectada como fechada;
- falso positivo: olhos abertos são detectados como fechados e boca fechada é detectada como aberta
- falso negativo: olhos abertos são detectados como abertos e boca fechada é detectada como fechada.

Tabela 2 – Matriz de Confusão de 320 testes

		Classe verdadeira	
		Positivo	Falso
Classe esperada	Positivo	263 (verdadeiro positivo)	44 (verdadeiro negativo)
	Falso	12 (falso positivo)	0 (falso negativo)

Fonte: autor.

De acordo com a Matriz de Confusão global apresentada na Tabela 2, o sistema proposto obteve uma eficácia de 82,2%, obtido por meio da Equação 1, vista na Subseção 3.8.

6 CONSIDERAÇÕES FINAIS

Através do estudo teórico e levantamento bibliográfico realizado, foi possível desenvolver a proposta de um dispositivo embarcado para detecção de sonolência e desatenção ao dirigir em tempo real. O sistema permite alertar o motorista por meio de um alarme sonoro e visual, caso o estado de sonolência seja detectado. Na Seção 4 foram discutidos os testes e resultados do protótipo junto à sua eficiência. Diante dos resultados obtidos, pode-se dizer que o projeto possui uma taxa de eficiência de 82,2% e, de acordo com os trabalhos científicos de Noronha (2019) e Paranhos (2019), o protótipo possui eficiência equivalente.

Os desafios enfrentados para realização do trabalho foram a dificuldade da detecção em ambientes com pouca luz e a reflexão da luz IR nos óculos de grau. Outro ponto relevante é o de refinar o algoritmo com o intuito de realizar a detecção em tempo real, devido à capacidade de processamento do microcontrolador. Mesmo estabelecendo um parâmetro mais sensível para o número de frames por segundo para detecção, percebeu-se um *delay* na resposta do dispositivo.

Para trabalhos futuros, com o objetivo de melhorar a eficiência do algoritmo, pode-se implementar o código/hardware em um dispositivo FPGA. Além disso, o sistema de iluminação infravermelho pode ser melhor distribuído dentro do veículo e, através de testes, verificar a possibilidade de aumentar a eficiência.

A visão computacional é um dos campos que mais tem progredido na área da tecnologia (ROQUE, 2013). Tal área da tecnologia viabiliza o desenvolvimento sistemas de detecção de sonolência viáveis e de baixo custo, como o protótipo proposto.

REFERÊNCIAS

AZEVEDO, Adriel Lima; CORREA, Anderson Luiz. **Visão Computacional aplicada a Manipulador Robótico**. 2019. 97 f. TCC (Graduação) - Curso de Engenharia Elétrica,

Departamento Acadêmico de Eletrotécnica, Universidade Tecnológica Federal do Paraná, Curitiba, 2019.

BACKES, André Ricardo; SÁ JUNIOR, Jarbas Joaci de Mesquita. Introdução: o que é visão computacional? In: BACKES, André Ricardo; SÁ JUNIOR, Jarbas Joaci de Mesquita. **Introdução a Visão Computacional Usando MATLAB**. Rio de Janeiro: Alta Books, 2019. Cap. 1. p. 0-288. (978-85-508-0628-0).

BALBINOT, Amanda B.; ZARO, Milton A.; TIMM, Maria I. Funções psicológicas e cognitivas presentes no ato de dirigir e sua importância para os motoristas no trânsito: psychological and cognitive functions in the act of driving and its importance for drivers in traffic. **Ciências e Cognição**, Rio de Janeiro, v. 16, n. 2, p. 13-29, 22 jul. 2011.

CARVALHO JUNIOR, Jose Regivaldo de. **Processamento Digital de Imagens para a Identificação Automática de Falhas em Rolos dos Transportadores de Correias**. 109 f. Dissertação (Mestrado) - Curso de Engenharia de Controle e Automação, Instituto Tecnológico Vale, Universidade Federal de Ouro Preto, Ouro Preto, 2018.

DINIZ, Fábio Abrantes et al. RedFace: **Um Sistema de Reconhecimento Facial para Identificação de Estudantes em um Ambiente Virtual de Aprendizagem**. Revista Novas Tecnologias na Educação, Porto Alegre, v. 10, n. 3, p. 1-11, dez. 2012. Disponível em: <https://seer.ufrgs.br/index.php/renote/article/view/36403/23510>. Acesso em: 20 nov. 2021.

DNIT. **Departamento Nacional de Infraestrutura de Transportes - Impacto dos Acidentes**. Disponível em: <https://www.gov.br/infraestrutura/pt-br/assuntos/noticias/ultimas-noticias/codigo-de-transito-brasileiro-e-a-principal-ferramenta-para-se-evitar-acidentes> . Acesso em: 20 abr. 2021.

DOI. **Sistema embarcado para detecção e alerta de atitudes de desatenção ao dirigir**. Disponível em: <https://github.com/ruivomari/TCC>. (DOI 10.5281/zenodo.5748515).

FREUND, Yoav; SCHAPIRE, Robert E.. **A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting**. Journal Of Computer And System Sciences. New Jersey, p. 119-139. ago. 1997. Disponível em: <https://www.sciencedirect.com/science/article/pii/S002200009791504X>. Acesso em: 20 nov. 2021.

GLASENAPP, Luiz Alfonso; HOPPE, Aurélio Faustino. **VI: Dispositivo para identificação de veículos com documentação irregular**. 2019. 21 f. TCC (Graduação) - Curso de Ciência da Computação, Departamento de Sistemas e Computação, Universidade Regional de Blumenau, Blumenau, 2019. Disponível em: http://dsc.inf.furb.br/arquivos/tccs/monografias/2019_1_luiz-alfonso-glasenapp_monografia.pdf. Acesso em: 01 dez. 2021.

GUNAWAN, Teddy Surya et al. **Development of Face Recognition on Raspberry Pi for Security Enhancement of Smart Home System**. Indonesian Journal Of Electrical Engineering And Informatics. Indonesian, dez. 2017. p. 317-325. Disponível em: <http://section.iaesonline.com/index.php/IJEEI/article/view/361/205>. Acesso em: 01 dez. 2021.

GT. **Georgia Tech – Face Database**. Disponível em: http://www.anefian.com/research/face_reco.htm. Acesso em: 17 maio 2021.

INFORMESERVICE. **Informeservice - RC-SF 1 | Sensor de Fadiga**. Disponível em: <https://www.informeservice.com.br/produtos/rc-sf1-sensor-de-fadiga/>. Acesso em: 08 maio 2021.

MARENGONI, Maurício; STRINGHINI, Denise. Tutorial: Introdução à Visão Computacional usando OpenCV. **Revista de Informática Teórica e Aplicada**, Porto Alegre, v. 16, n. 1, p. 125-160, mar. 2010.

MARPRESS. **Marpres - Sensor de Fadiga**. Disponível em: <http://marpressbrasil.com.br/project/sensor-de-fadiga/>. Acesso em: 06 maio 2021.

MARWEDEL, Peter. **Embedded System Design: embedded systems foundations of cyber-physical systems, and the internet of things**. 4. ed. Germany: Springer, 2021. 445 p

MATAI, Janarbek; IRTURK, Ali; KASTNER, Ryan. **Design and Implementation of an FPGA-based Real-Time Face Recognition System**. In: IEEE INTERNATIONAL SYMPOSIUM ON FIELD-PROGRAMMABLE CUSTOM COMPUTING MACHINES, 19., 2011, La Jolla. Symposium. California: Ieee, 2011. p. 97-100. Disponível em: <https://ieeexplore.ieee.org/abstract/document/5771257>. Acesso em: 01 dez. 2021.

NHTSA. **National Highway Traffic Safety Administration – Drowsy Driving**. Disponível em: <https://www.nhtsa.gov/risky-driving/drowsy-driving>. Acesso em: 25 maio 2021.

NORONHA, Fernando de Almeida *et al.* **Deteção de fadiga a partir da análise de imagens faciais**. Colloquium Exactarum, Presidente Prudente, v. 11, n. 2, p. 34-45, 31 jul. 2019. Disponível em: <https://revistas.unoeste.br/index.php/ce/article/view/3168/2794>. Acesso em: 20 abr. 2021.

OPENCV. **OpenCV - Hough Circle Transform**. Disponível em: https://docs.opencv.org/3.4/d4/d70/tutorial_hough_circle.html. Acesso em: 11 ago. 2021.

OPENCV2. **OpenCV - Morphological Transformations**. Disponível em: https://docs.opencv.org/3.4/d9/d61/tutorial_py_morphological_ops.html. Acesso em: 20 nov. 2021.

PARANHOS, Vinícius Kerber. **Sistema de Detecção de Sonolência**. 2019. 67 f. TCC (Graduação) - Curso de Engenharia Elétrica, Faculdade de Engenharia e Arquitetura, Universidade de Passo Fundo, Passo Fundo, 2019. Disponível em: <http://repositorio.upf.br/bitstream/riupf/1709/1/PF2019Vinicius%20Kerber%20Paranhos.pdf>. Acesso em: 12 nov. 2021.

PYSCIENCE. **Numpy**. Disponível em: <http://pyscience-brasil.wikidot.com/module:numpy>. Acesso em: 20 nov. 2021.

PYIMAGESEARCH. **Drowsiness detection with OpenCV**. Disponível em: <https://www.pyimagesearch.com/2017/05/08/drowsiness-detection-opencv/>. Acesso em: 25 ago. 2021.

QUEIROZ, Kédson Lopes de. **Sistema Baseado em Vídeo para Detecção de Sonolência em Motoristas**. 2011. 61 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Faculdade de Tecnologia, Universidade de Brasília, Brasília, 2011. Disponível em: <https://repositorio.unb.br/handle/10482/10704>. Acesso em: 01 maio 2021

RICHARDSON, Matt; WALLACE, Shawn. **Primeiros Passos com o Raspberry Pi**. São Paulo: Novatec Editora Ltda., 2013. 36 p. (978-85-7522-345-1). Disponível em: <https://s3.novatec.com.br/capitulos/capitulo-9788575223451.pdf>. Acesso em: 20 nov. 2021.

RODRIGUES, Matheus Bezerra Estrela. **Estudo da aplicação do algoritmo Viola-Jones à detecção de pneus com vistas ao reconhecimento de automóveis**. 2012. 69 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Centro de Engenharia Elétrica e Informática,

Universidade Federal de Campina Grande, Campina Grande, 2012. Disponível em: <http://dspace.sti.ufcg.edu.br:8080/jspui/bitstream/riufcg/1861/1/MATHEUS%20BEZERRA%20ESTRELA%20RODRIGUES%20-%20DISSERTA%C3%87%C3%83O%20PPGCC%202012..pdf>. Acesso em: 20 nov. 2021.

ROQUE, Pedro Miguel da Silva. **Técnicas de Visão Computacional para a detecção automática de padrões de fadiga**. 2013. 57 f. Dissertação (Mestrado) - Curso de Engenharia Informática, Departamento de Informática, Universidade de Évora, Évora, 2013.

SCIPY. **Scipy**. Disponível em: <https://scipy.org/>. Acesso em: 20 nov. 2021.

SILVA, Julio Batista. **Explorando o algoritmo de Viola-Jones na detecção e reconhecimento facial**. 2018. 91 f. TCC (Graduação) - Curso de Engenharia de Computação, Departamento de Computação, Universidade Federal de São Carlos, São Carlos, 2018. Disponível em: https://juliobs.com/Julio_Batista_Silva-TCC-Face_Recognition.pdf. Acesso em: 20 nov. 2021.

SILVA, José Ivson Soares da. **Reconhecimento Facial em Imagens de Baixa Resolução**. 2015. 77 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Centro de Informática, Universidade Federal de Pernambuco, Recife, 2015.

SOUZA, Wallace Bruno Silva de. **Detecção e Reconhecimento Automáticos de Sinais de Trânsito**. 2015. 81 f. TCC (Graduação) - Curso de Ciências da Computação, Instituto de Ciências Exatas, Universidade de Brasília, Brasília, 2015.

Tecnologia GPS. **Tecnologia GPS - Detector de fadiga e distração do motorista DCM-01**. Disponível em: <https://www.tecnologiagps.com.br/produtos/dcm-01>. Acesso em: 08 maio 2021.

UFOP. **FPGA – A Flexibilidade No Projeto De Hardware. Parte 1 De 3**. Disponível em: <http://www2.decom.ufop.br/imobilis/fpga-o-prodigio-de-flexibilidade/#:~:text=Defini%C3%A7%C3%A3o%20do%20FPGA,uso%20de%20v%C3%A1rios%20projetos%20eletr%C3%B4nicos>. Acesso em: 19 maio 2021.

VW. **Volkswagen - Segurança**. Disponível em: <https://www.vw.com.br/pt/volkswagen/tecnologia/seguranca.html>. Acesso em: 20 abr. 2021.

WHO. WORLD HEALTH ORGANIZATION. (org.). **Global Status Report on Road Safety 2018**. Geneva: Who, 2018. 403 p. Disponível em: <https://www.unifor.br/documents/929808/930330/9789241565684-eng.pdf/36966e12-d0f4-725a-2f4a-a7ed90a54855>. Acesso em: 03 abr. 2021.

AGRADECIMENTOS

Aos meus pais, Nilma e Paulo, a minha irmã, Jaqueline, e a minha família, que nortearam minha formação enquanto indivíduo e me deram toda a base e suporte para a minha formação enquanto profissional.

Ao meu namorado, Leonardo, pela contribuição no desenvolvimento do projeto, incentivo e carinho nesse ano.

Aos meus amigos e colegas de faculdade pelo apoio e motivação diária nessa jornada.

E a todos que se fizeram presente neste processo.