

**UNIVERSIDADE DE CAXIAS DO SUL
ÁREA DO CONHECIMENTO DE CIÊNCIAS EXATAS E
ENGENHARIAS**

SAMUEL FORMIGHERI

**DETECÇÃO DE SONOLÊNCIA EM MOTORISTAS UTILIZANDO
PROCESSAMENTO DE IMAGENS**

CAXIAS DO SUL

2022

SAMUEL FORMIGHERI

**DETECÇÃO DE SONOLÊNCIA EM MOTORISTAS UTILIZANDO
PROCESSAMENTO DE IMAGENS**

Trabalho de Conclusão de Curso
apresentado como requisito parcial
à obtenção do título de Bacharel em
Engenharia de Computação na Área
do Conhecimento de Ciências Exatas e
Engenharias da Universidade de Caxias
do Sul.

Orientador: Prof. Dr. André Luis
Martinotto

CAXIAS DO SUL

2022

SAMUEL FORMIGHERI

**DETECÇÃO DE SONOLÊNCIA EM MOTORISTAS UTILIZANDO
PROCESSAMENTO DE IMAGENS**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel em Engenharia de Computação na Área do Conhecimento de Ciências Exatas e Engenharias da Universidade de Caxias do Sul.

Aprovado em 00/11/2022

BANCA EXAMINADORA

Prof. Dr. André Luis Martinotto
Universidade de Caxias do Sul - UCS

Prof. Dr. Guilherme Holsbach Costa
Universidade de Caxias do Sul - UCS

Prof. Dr. Ricardo Vargas Dorneles
Universidade de Caxias do Sul - UCS

*Este trabalho é dedicado às crianças adultas
que, quando pequenas, sonharam em se tornar
cientistas.*

AGRADECIMENTOS

Agradeço a minha mãe, Marlene Pistor Formigheri, que sempre me incentivou a prosseguir e apoiou independentemente da situação em todos estes anos de universidade. Ao meu pai, Gelson Formigheri, que sempre foi suporte e apoio em toda minha caminhada.

Ao meu orientador professor André, por todo conhecimento transmitido e empenho dedicado. E aos amigos e colegas de graduação, que fiz durante esses anos.

“A necessidade é a mãe da inovação.”

Platão

RESUMO

A sonolência e a fadiga dos motoristas estão entre as maiores causas de acidentes rodoviários graves. Neste sentido, as empresas do setor automobilístico investem fortemente em pesquisas e no desenvolvimento de soluções para a detecção da sonolência em motoristas. Porém, frequentemente, essas soluções utilizam-se de sensores caros ficando restritas aos carros topo de linha. Assim, neste trabalho foi desenvolvido um sistema de baixo custo para a detecção de sonolência em motoristas. Esse é um sistema não intrusivo baseado na utilização de um *smartphone* para a captura de imagens do motorista. Após captura, é realizado um pré-processamento da imagem onde é utilizado o método de interpolação pelo vizinho mais próximo para uma redução das dimensões da mesma. A detecção da face é realizada através da utilização do Classificador *Haar*. O reconhecimento dos olhos e a classificação (abertos ou fechados) é realizada através de uma rede neural convolucional do tipo *BlazeFace*. Por fim, a definição do estado de sonolência é realizada através da métrica de percentual de fechamento dos olhos esquerdo e direito (*PERCLOS - Percentage of Closure*). O sistema desenvolvido mostrou-se viável, apresentando resultados satisfatórios, com um baixo tempo de execução. Esse apresentou uma precisão média de 98,61% em ambientes com iluminação normal e sem obstáculos ocultando mais de uma característica da face. O tempo total de processamento de um *frame* foi de aproximadamente 343,5 ms em um *Galaxy M20* e de 429 ms em *iPhone 8*, sendo assim, o sistema é capaz de analisar de 2 a 3 *frames* por segundo em um dispositivo intermediário.

Palavras-chave: Detecção de Sonolência. Smartphone. Classificador *Haar*. *MobileNet*. *SSD*. *BlazeFace*. *PERCLOS - Percentage of Closure*.

LISTA DE FIGURAS

Figura 1 – Equalização de histograma com limitação de contraste	25
Figura 2 – Filtro de Sobel	26
Figura 3 – Correção geométrica	26
Figura 4 – Região de máxima do sinal de simetria	28
Figura 5 – Imagem dos olhos extraída ao redor da região de máxima do sinal de simetria	29
Figura 6 – Transformada de <i>Hough</i>	30
Figura 7 – Fluxograma com as etapas da proposta de solução	32
Figura 8 – Posição do <i>smartphone</i> para captura da imagem facial	33
Figura 9 – Imagem após conversão para escala cinza	34
Figura 10 – Recursos de <i>Haar</i> selecionados após a etapa de treinamento	35
Figura 11 – Recursos de <i>Haar</i> : a soma dos <i>pixels</i> dos retângulos brancos é subtraída da soma dos <i>pixels</i> dos retângulos cinza	36
Figura 12 – Exemplo de aplicação da imagem integral	36
Figura 13 – Representação da imagem de entrada em um tamanho de 5×7 <i>pixels</i> com a intensidade de cada pixel, ao lado, sua imagem integral	37
Figura 14 – Representação esquemática de uma cascata de detecção	38
Figura 15 – Imagem após a aplicação do método <i>Classificador Haar</i>	38
Figura 16 – Mapa de características F e kernel de convolução K	39
Figura 17 – <i>Kernels</i> de convolução em profundidade	40
Figura 18 – Filtros convolucionais 1×1 chamados de filtros de convolução pontual (<i>Pointwise</i>) no contexto da convolução separável em profundidade	40
Figura 19 – Camadas da rede neural convolucional <i>MobileNet</i>	41
Figura 20 – Imagem do rosto após a aplicação do método de <i>BlazeFace</i> . Os 6 pontos de referência facial estão destacados em azul	43
Figura 21 – Fatia de mercado de cada um dos sistemas operacionais para dispositivos móveis (Jun 2022)	45
Figura 22 – Telas iniciais do aplicativo	47
Figura 23 – Telas principais do aplicativo	47
Figura 24 – Telas de detecção da face e classificação	48
Figura 25 – Mensagem de alerta	49
Figura 26 – Estrutura para simular ambiente de alta luminosidade	50
Figura 27 – Imagem capturada com alta luminosidade	51
Figura 28 – Imagem capturada com baixa luminosidade	52
Figura 29 – Motorista com óculos	52
Figura 30 – Imagem capturada com uma inclinação de 45°	53
Figura 31 – Exemplo de imagem capturada em que o rosto está de cabeça para baixo	54

Figura 32 – Exemplo de imagem capturada em testes com a mão encobrindo a boca . . .	54
Figura 33 – Exemplo de imagem capturada em testes com a mão encobrindo regiões da face	55

LISTA DE TABELAS

Tabela 1 – <i>MobileNet</i> vs outros modelos	42
Tabela 2 – Etapas e seus desempenhos em cada uma das plataformas	50

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
ADAS	<i>Advanced driver assistance systems</i>
AVECLOS	<i>Average of eye closure</i>
CLAHE	<i>Contrast Limited Adaptive Histogram Equalization</i>
ECG	Electrocardiograma
EEG	Eletroencefalograma
EMG	Eletromiograma
EoG	Eletrooculograma
GPS	<i>Global Positioning System</i>
HF	<i>High Frequency</i>
HRV	<i>Heart Rate Variability</i>
LF	<i>Low Frequency</i>
NHTSA	<i>National Highway Traffic Safety Administration</i>
NSF	<i>National Sleep Foundation</i>
OMS	Organização Mundial da Saúde
PERCLOS	<i>Percentage of Closure</i>
REM	<i>Rapid eye movements</i>
SSD	<i>Single Shot Multibox Detector</i>

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVOS	15
1.2	ESTRUTURA DO TRABALHO	15
2	DETECÇÃO DO ESTADO DE SONOLÊNCIA	17
2.1	Medidas baseadas no veículo	17
2.1.1	Posição da pista	17
2.1.2	Movimento do volante	18
2.1.3	Pressão nos pedais	18
2.2	Medidas baseadas no comportamento fisiológico	18
2.2.1	Eletroencefalograma	19
2.2.2	Eletrocardiograma	20
2.2.3	Eletrooculograma	20
2.3	Medidas baseadas no comportamento do motorista	21
2.3.1	Características oculares	21
2.3.2	Postura da cabeça	22
2.3.3	Identificação do Bocejo	23
3	DETECÇÃO DO ESTADO DE SONOLÊNCIA ATRAVÉS DO PROCESSAMENTO DE IMAGENS	24
3.1	Captura e Pré-processamento das Imagens	24
3.2	Detecção da Face	27
3.3	Detecção dos olhos e da boca	28
3.4	Detecção do estado de sonolência do motorista	30
4	DESCRIÇÃO DO SISTEMA DESENVOLVIDO	32
4.1	Captura e Pré-processamento das Imagens	33
4.2	Detecção da Face	34
4.3	Reconhecimento dos olhos	39
4.4	Detecção do estado de sonolência	44
5	IMPLEMENTAÇÃO DESENVOLVIDA E RESULTADOS OBTIDOS	45
5.1	Interface do Aplicativo Desenvolvido	46
5.2	Testes nas plataformas Android e iOS	48
5.3	Testes de Desempenho	49
5.4	Testes em Ambientes com Alta Luminosidade	50
5.5	Testes em Ambientes com Baixa Luminosidade	51

5.6	Motorista que utiliza óculos	52
5.7	Mal posicionamento da câmera	53
5.8	Parte da face encoberta	53
6	CONSIDERAÇÕES FINAIS	56
6.1	Trabalhos futuros	57
	REFERÊNCIAS	58

1 INTRODUÇÃO

O transporte, seja de cargas ou pessoas, é uma atividade fundamental para a sociedade. Essa atividade está diretamente relacionada com a economia de um país, uma vez que quanto maior for o número de indústrias, comércios e atividades econômicas, maior será a demanda pelos meios de transporte. Por isso o transporte é uma atividade tão importante, seja para satisfazer as necessidades dos indivíduos ou mesmo para o desenvolvimento da sociedade. No entanto, quanto maior for a demanda pelo transporte, aumenta-se também o tráfego de veículos e, conseqüentemente, o número de acidentes de trânsito. De acordo com a Organização Mundial da Saúde (OMS), 1,35 milhões de pessoas morrem a cada ano por causa de acidentes rodoviários (OMS, 2021).

Os acidentes podem ocorrer por diversos fatores, sejam eles por falhas mecânicas ou por falhas humanas, causando diversos prejuízos à sociedade. Segundo WELLS (2007) é amplamente conhecido que a falha de um determinado componente ou sistema em um carro raramente é a origem de um acidente fatal, pelo menos nos países onde a frota de carros é relativamente nova, e o regime de inspeção e reparo de veículos é relativamente robusto. Nestes países, a principal origem de acidentes está no motorista, sendo necessária uma compreensão cultural para encontrar soluções para os problemas de transporte (WELLS, 2007).

Segundo a OMS (2021), mais de 90% das mortes no trânsito ocorrem em países de baixa e média renda, sendo que as taxas de mortalidade por lesões causadas pelo trânsito são mais altas em países africanos. Mesmo em países de alta renda, as pessoas de níveis socioeconômicos mais baixos são as mais propensas a se envolver em acidentes de trânsito. Ainda, segundo a OMS (2021), as lesões causadas pelo trânsito são a principal causa de morte de crianças e adultos entre 5 a 29 anos. Além disso, os homens são mais propensos a se envolver em acidentes de trânsito do que as mulheres. De fato, cerca de 73% de todas as mortes no trânsito ocorrem entre jovens do sexo masculino com idade inferior a 25 anos. Assim, estatisticamente, os jovens do sexo masculino têm quase 3 vezes mais chances de morrer em um acidente de trânsito do que as jovens do sexo feminino.

Ainda de acordo com a OMS (2021), vários são os fatores de risco que podem causar um acidente, sendo que os principais são: a alta velocidade, dirigir sob efeito de álcool e outras substâncias psicoativas, não uso de capacetes em motocicleta, cintos de segurança e sistemas de retenção para crianças, motoristas distraídos, infraestrutura rodoviária insegura, veículos inseguros, cuidados inadequados após o acidente, aplicação inadequada das leis de trânsito e condições físicas e fisiológicas inadequadas dos motoristas.

De acordo com a *NHTSA (National Highway Traffic Safety Administration - Administração Nacional de Segurança do Trânsito)* cerca de 100 mil acidentes ocorrem anualmente nos Estados Unidos por causa da sonolência do motorista. No ano de 2009, a Fundação Nacional do Sono dos Estados Unidos (*NSF* do inglês *National Sleep Foundation*), relatou que 54% dos motoristas adultos dirigiram um veículo enquanto se sentiam sonolentos e 28% deles adormeceram na direção (SAHAYADHAS; SUNDARAJ; MURUGAPPAN, 2012).

No Brasil, segundo a *ABRAMET (2020)*, a terceira condição de saúde que mais causou acidentes fatais entre 2014 e 2020 foi o sono. No estudo realizado pela *ABRAMET (Associação Brasileira de Medicina de Tráfego)* em conjunto com a Polícia Rodoviária Federal, verificou-se que, entre janeiro de 2014 e julho de 2020, foram causados pela sonolência dos motoristas 22.683 acidentes (uma média de 9,56 acidentes por dia). Como consequências desses acidentes foram registrados 2.092 mortos e 22.645 feridos.

A sonolência excessiva acarreta uma redução do tempo de reação, julgamento, visão e dificuldades no processo da informação e memória de curto prazo. Isso implica na perda do controle do veículo pelo motorista, uma ação que muitas vezes resulta na colisão com outro veículo ou com objetos estacionários (*ABRAMET, 2020*). Destaca-se que todos os motoristas podem ser vítimas de sonolência durante a condução, seja após uma noite de sono muito curta, por uma condição fisiológica alterada ou durante longas viagens (*ALSHAQAQI et al., 2013*).

A pesquisa sobre o reconhecimento da sonolência é uma área ativa e importante para uma melhoria da segurança no trânsito, podendo evitar acidentes causados pelo sono. Assim, é importante a utilização de novas tecnologias para projetar e construir sistemas robustos e confiáveis, capazes de monitorar os motoristas e medir seu nível de atenção durante todo o processo de condução (*ALSHAQAQI et al., 2013*).

Nos últimos anos, câmeras instaladas em veículos se tornaram comuns para efeitos de segurança e registro de acidentes. Em alguns países, como na Rússia, são utilizadas as imagens de uma câmera dianteira como prova para o ressarcimento do seguro ou para a defesa no caso de uma multa indevidamente aplicada. Além disso, os *smartphones* estão sendo cada vez mais utilizados em suportes no painel dianteiro do veículo, dando a possibilidade de utilização de um grande número de sensores, incluindo áudio, gravadores, sistemas de posicionamento global (GPS do inglês *Global Positioning System*), acelerômetros, sensores de luz e temperatura (*KWAPISZ; WEISS; MOORE, 2011*).

Aliado à crescente utilização da câmera dianteira e do *smartphone* nos veículos, o campo de visão computacional avançou significativamente nos últimos anos. Com isso, pode-se aplicar sistemas de reconhecimento de expressão facial, já que o estado de sonolência provoca mudanças físicas e fisiológicas no condutor. Ao analisar o comportamento facial do motorista e os sinais físicos e fisiológicos que involuntariamente ocorrem no estado de sonolência (e em alguns casos até instantes anteriores), pode-se detectar e alertar o motorista evitando que ele entre em sono profundo (*VURAL et al., 2010*).

Dentro deste contexto, neste trabalho foi desenvolvido um protótipo de um sistema de baixo custo para a detecção de um estado de sonolência do motorista utilizando dispositivos móveis (*smartphone*) e técnicas de reconhecimento facial. Apesar de já existirem vários sistemas para a detecção da sonolência, esses geralmente recorrem a equipamentos de alto custo. Para a detecção da sonolência foi realizada uma análise das características oculares do motorista. A partir da análise dessas características é possível monitorar e disparar um alarme sonoro em caso de detecção da sonolência.

1.1 OBJETIVOS

Este trabalho apresentou como principal objetivo desenvolver um sistema de baixo custo para o reconhecimento de sonolência através da análise de imagens faciais do motorista. Para que o objetivo principal deste trabalho fosse atingido os seguintes objetivos específicos foram realizados:

1. Construção de uma base de dados de imagens faciais;
2. Identificação e categorização dos traços faciais ligados a sonolência;
3. Identificação das características e dos métodos mais utilizados para o reconhecimento de sonolência através de imagens faciais;
4. Desenvolvimento de uma proposta de solução para um sistema de reconhecimento de sonolência de motoristas;
5. Implementação de um protótipo para o reconhecimento de sonolência de motoristas utilizando imagens capturadas através de um *smartphone*.
6. Realização dos testes e a análise dos resultados obtidos.

1.2 ESTRUTURA DO TRABALHO

O presente trabalho se divide em seis capítulos, organizados da seguinte forma:

- Neste capítulo foi realizada uma breve introdução do trabalho, apresentando sua motivação e os seus principais objetivos referentes ao tema.
- No Capítulo 2 é realizada uma apresentação dos principais tipos de sistemas para a detecção de sonolência em motoristas. Nele é realizada uma breve descrição desses sistemas e das principais técnicas utilizadas.
- O Capítulo 3 apresenta as principais técnicas e algoritmos que são utilizados para a detecção do sono através do processamento de imagens. Além disso, são apresentados trabalhos relacionados e relevantes dentro desta área.

- O Capítulo 4 descreve a proposta de solução desenvolvida, apresentando as técnicas e os métodos utilizados.
- O Capítulo 5 descreve os detalhes de implementação da solução e apresenta os principais resultados obtidos.
- Por fim, no Capítulo 6 são apresentadas as considerações finais, bem como sugestões de trabalhos futuros.

2 DETECÇÃO DO ESTADO DE SONOLÊNCIA

Segundo UENO; KANEDA; TSUKINO (1994), pode-se fazer uma divisão dos sistemas para a detecção de sonolência em dois grupos: os intrusivos e os não intrusivos. Os sistemas intrusivos se caracterizam pela necessidade do usuário utilizar sensores acoplados, que tendem a ser desconfortáveis. Como exemplo, pode-se citar os sistemas que medem a pressão arterial (HAYASHI *et al.*, 2005), sistemas de análise de gráficos de eletroencefalografia (JOHN; RISSER; KOBUS, 2006; PARIKH; MICHELI-TZANAKOU, 2004) e sistemas de análise de frequência cardíaca. Já os sistemas não intrusivos não necessitam de sensores acoplados ao condutor, sendo mais confortáveis para o motorista. Os métodos não intrusivos comumente utilizados são baseados em sistemas de visão computacional (ERIKSSON; PAPANIKOTOPOULOS, 1997). Além disso, segundo SAHAYADHAS; SUNDARAJ; MURUGAPPAN (2012), as formas de monitoramento da sonolência do motorista podem ser divididas em três categorias, que utilizam medidas baseadas: no veículo, no comportamento do motorista e no comportamento fisiológico do motorista.

2.1 MEDIDAS BASEADAS NO VEÍCULO

As medidas baseadas no veículo são não intrusivas, já que não necessitam de nenhum tipo de sensor acoplado ao motorista. Várias métricas podem ser analisadas, sendo que as mais comuns são o desvio de posição da pista, o movimento do volante e a pressão do pedal do acelerador. Neste tipo de sistema existe um monitoramento constante, para que qualquer valor que ultrapasse um limite especificado seja identificado como sinal da sonolência do motorista (LIU; HOSKING; LENNÉ, 2009; FORSMAN *et al.*, 2013).

2.1.1 Posição da pista

Esse tipo de detecção baseia-se no fato de que o condutor sonolento, ou mesmo desatento, costuma desviar a posição do veículo ao dirigir. Assim, o objetivo desses sistemas é avisar o condutor sempre que for identificada uma saída não intencional da pista. Esses sistemas geralmente são baseados no reconhecimento de imagens e utilizam as marcações das estradas. Porém, eles também podem utilizar radares ou sistemas de posicionamento global (GPS do inglês *Global Positioning System*) (CHACON-MURGUÍA; PRIETO-RESENDIZ, 2015).

A principal limitação deste tipo de sistema é que ele é dependente de fatores externos como, por exemplo, a sinalização da estrada. De fato, no caso de métodos que se baseiam nas linhas da rodovia, as condições das linhas, condições climáticas e a iluminação podem afetar significativamente o desempenho do sistema (SAHAYADHAS; SUNDARAJ; MURUGAPPAN, 2012).

Entre os sistemas que fazem uso dessa técnica, pode-se citar: o sensor de visão, desenvolvido pela empresa *Mobileye NV* (EDWARDS *et al.*, 2007); o aviso de saída de pista adaptável, desenvolvido pelo *Royal Institute of Technology* e patrocinado pela Scania (DERAM, 2004); e o aviso de saída de pista, desenvolvido pela *General Motors*.

2.1.2 Movimento do volante

Esse tipo de sistema baseia-se no monitoramento das reações do condutor através de um acompanhamento dos movimentos do volante. Usando um sensor de ângulo, montado na coluna de direção, o comportamento de direção do motorista é verificado. Quando o motorista está em um estado de sonolência, ocorre uma redução no número de micro correções em comparação com a condução normal (FENG; ZHANG; CHENG, 2009). Assim, considerando apenas os pequenos movimentos do volante (ângulos entre 0,5° e 5°), que são necessários para ajustar a posição do veículo na pista, é possível determinar o estado de sonolência do motorista e assim fornecer uma alerta se necessário (OTMANI *et al.*, 2005).

No entanto, conforme VURAL (2009), essa medida apresenta limitações significativas, decorrentes do fato que essa técnica necessita de ambientes particulares, sendo dependente das características geométricas da estrada e, em um menor grau, das características cinéticas do veículo.

2.1.3 Pressão nos pedais

No trabalho desenvolvido por KIM; KIM; HAHN (2008) é utilizada uma medida de pressão nos pedais do acelerador e do freio. Esse sistema baseia-se no fato de que um motorista distraído ou sonolento tende a utilizar o freio de forma mais repentina, ou acelerar de forma mais brusca. Além disso, um motorista desatento tende a exercer uma pressão maior nos pedais.

2.2 MEDIDAS BASEADAS NO COMPORTAMENTO FISIOLÓGICO

Esses sistemas são baseados em medidas intrusivas, já que necessitam de sensores acoplados ao condutor a fim de obter os seus sinais fisiológicos, sendo que os sinais mais utilizados são: eletrocardiograma (ECG), eletromiograma (EMG), eletroencefalograma (EEG) e eletro-oculograma (EoG). A medição dos sinais fisiológicos brutos é propensa a ruídos devido ao movimento que está envolvido com a condução. Assim, a fim de eliminar o ruído, várias técnicas, como filtros passa baixas e diferenciadores digitais têm sido utilizadas para remover ou reduzir o ruído do sinal (MICHAIL *et al.*, 2008).

A confiabilidade e precisão deste tipo de sistema é alta em comparação com outros métodos e sua principal vantagem está no fato de que os sinais fisiológicos começam a mudar em estágios iniciais de sonolência. Assim, esses sinais são adequados para detectar a sonolência com poucos falsos positivos, permitindo alertar o motorista em tempo hábil. Além disso, essas características fisiológicas não implicam somente em sonolência, podendo ser utilizadas para a identificação de outras condições, como por exemplo, a fadiga emocional ou física (ARAKAWA, 2021).

A natureza intrusiva desse tipo de sistema continua a ser uma questão a ser resolvida, uma vez que a colocação de sensores no motorista, pode ser um obstáculo para uma boa condução (SAHAYADHAS; SUNDARAJ; MURUGAPPAN, 2012). De forma a reduzir esse problema, alguns sistemas utilizam-se de dispositivos sem fio para medir os sinais fisiológicos de formas menos intrusivas. Por exemplo, em alguns sistemas, são colocados eletrodos no corpo do motorista e os sinais são obtidos utilizando tecnologias sem fio como *Zigbee* (KOBAYASHI, 2013) e *Bluetooth* (KLINGEBERG; SCHILLING, 2012). Em outros sistemas, os sinais fisiológicos são medidos de forma não intrusiva, a partir de eletrodos colocados no volante (YU, 2009; GÓMEZ-CLAPERS; CASANELLA, 2011) ou no banco do motorista (YU, 2009; BAEK *et al.*, 2011).

2.2.1 Eletroencefalograma

O eletroencefalograma (EEG) é uma medida fisiológica que detecta a atividade elétrica produzida pelo cérebro por meio de eletrodos acoplados a cabeça (DINGES *et al.*, 1998). Esse é o sinal fisiológico mais utilizado para medir o estado de sonolência por se tratar de um índice de neuroatividade direta (SAHAYADHAS; SUNDARAJ; MURUGAPPAN, 2012).

O sinal EEG apresenta várias bandas de frequência, partindo de frequências mais baixas, no estado de sono, até frequências mais altas, no estado de alerta. As bandas de frequências são comumente divididas em: banda *delta* (0.5 - 4 Hz) que corresponde a atividade do sono; a banda *theta* (4 - 8 Hz) que corresponde a sonolência; a banda *alpha* (8 - 13 Hz) que representa a sensação de relaxamento; e a banda *beta* (13 - 25 Hz) que corresponde à situação de alerta (AKIN *et al.*, 2008; LIN *et al.*, 2012).

No entanto, é difícil construir um sistema que use EEG para a detecção de sonolência do motorista uma vez que esse é suscetível a ruídos. Além disso, os eletrodos são conectados à cabeça e o movimento da cabeça torna-se restrito. Desta forma, esse tipo de sistema não é utilizado comercialmente, existindo apenas estudos acadêmicos (CHACON-MURGUIA; PRIETO-RESENDIZ, 2015). Como exemplo, pode-se citar o trabalho desenvolvido por DINGES *et al.* (1998), que se utiliza de ondas de EEG que são capturadas a cada 1,2 segundos na área occipital.

2.2.2 Eletrocardiograma

Outra métrica que é utilizada para a detecção de um estado de sonolência é a variabilidade dos batimentos cardíacos (*HRV - Heart Rate Variability*), que é medida por meio de um eletrocardiograma ou através do nível de pressão arterial. A *HRV* está relacionada com o sistema nervoso autônomo, por isso também pode ser utilizada para avaliar a carga de trabalho e a tensão emocional de um indivíduo (LIN *et al.*, 2006). Existem basicamente três índices que podem ser derivados dos dados de *HRV*: a atividade do nervo simpático, a atividade do nervo parassimpático e a própria variabilidade dos batimentos (HAYASHI *et al.*, 2005).

A detecção do estado de sonolência é geralmente realizada no *ECG* utilizando a proporção entre: *LF* (*Low Frequency* - baixa frequência), que representa a atividade do nervo simpático; e *HF* (*High Frequency* - alta frequência), que representa a atividade do nervo parassimpático. A relação *LF/HF*, contém informações relevantes sobre a fadiga, sendo que essa proporção diminui progressivamente à medida que o motorista passa de um estado de alerta para um estado de sonolência (YANG; LIN; BHATTACHARYA, 2010; PATEL *et al.*, 2011; BYEON *et al.*, 2006). Da mesma forma, um aumento na proporção *LF/HF* indica uma maior carga de trabalho mental ou estado de alerta (SATO *et al.*, 1998).

A privação do sono por si só, não influencia diretamente nos sinais cerebrais que controlam a sonolência, sendo que a duração da tarefa também tem uma forte influência. Pesquisadores da Universidade de Kanazawa (Japão), juntamente com a Honda, observaram que o nível dos batimentos diminuiu gradualmente até estabilizarem em uma baixa frequência, ao dirigir em um ambiente monótono, e esta etapa foi seguida por um estado de sonolência. Os batimentos diminuem, pois o ambiente monótono implica uma situação entediante e, conseqüentemente, facilita a sonolência (YAMAKOSHI *et al.*, 2006).

2.2.3 Eletrooculograma

Alguns sistemas utilizam um eletrooculograma para a identificação da sonolência, ou seja, a sonolência é identificada a partir da análise do movimento ocular (KHUSHABA *et al.*, 2010; HU; ZHENG, 2009; KURT *et al.*, 2009). O método do eletrooculograma baseia-se na diferença de potencial elétrico existente entre a córnea e a retina. Essa diferença de potencial gera um campo elétrico que reflete a orientação dos olhos, sendo que este campo elétrico é o sinal medido pelo eletrooculograma (*EOG*).

No trabalho desenvolvido por HU; ZHENG (2009) foi analisado o movimento horizontal dos olhos, através de um eletrodo de prata (Ag) e cloro (Cl) posicionado no canto externo de cada olho e um terceiro eletrodo de referência que foi posicionado no centro da testa do motorista. Esses eletrodos foram posicionados de forma a possibilitar a detecção dos movimentos *REM* (*Rapid eye movements* - Movimentos rápidos dos olhos), que ocorrem quando se está sonolento.

2.3 MEDIDAS BASEADAS NO COMPORTAMENTO DO MOTORISTA

Uma pessoa sonolenta apresenta uma série de movimentos faciais característicos, incluindo movimentos rápidos dos olhos e piscar constante, além de balançar a cabeça e bocejar frequentemente. Assim, essas características podem ser utilizadas para a detecção do estado de sonolência. Estes sistemas frequentemente utilizam-se de imagens, sendo considerados mais adequados, pois não são intrusivos (JI; ZHU; LAN, 2004). A principal limitação deste tipo de sistema é a iluminação, uma vez que as câmeras normais não possuem uma boa captura de imagens à noite (BERGASA *et al.*, 2006).

Diversos trabalhos desenvolvidos que realizam a análise do comportamento do motorista através de imagens, baseiam-se no piscar dos olhos (BERGASA *et al.*, 2006). Como exemplo, pode-se citar o sistema de monitoramento de condução da *Toyota*, que se utiliza da direção dos olhos e da abertura/fechamento das pálpebras para a detecção de um estado de sonolência. Esse sistema apresenta ainda uma funcionalidade que permite parar o carro se a postura de condução do motorista estiver distorcida ou se o condutor não responder aos avisos (AISIN, 2021).

2.3.1 Características oculares

O piscar dos olhos é influenciado por fatores psicológicos e comportamentais, fornecendo informações sobre a capacidade perceptiva e cognitiva de um indivíduo. De fato, a frequência do piscar dos olhos é diretamente afetada pela fadiga, lesões oculares, medicamentos e doenças (TABRIZI; ZOROOFI, 2009).

Os olhos cansados são mais suscetíveis a altas taxas de piscar, sendo que o número de piscadas aumenta quando o indivíduo começa a se sentir sonolento (partindo de um estado de vigília completa). No entanto, o número de piscadas diminui quando o sujeito fica mais sonolento após o nível de vigília estar menor (TANAKA, 1999). A resposta da pupila também encontra-se associada com a variabilidade do batimento cardíaco (HRV), já que a pupila é regulada pelo sistema nervoso autônomo. Desta forma, as dilatações e contrações da pupila também podem fornecer informações sobre o estado de vigília ou sonolência do indivíduo (MURATA; IWASE, 1998).

Os sistemas baseados em características oculares geralmente envolvem as etapas de detecção e extração da face, detecção e rastreamento do rosto, detecção e rastreamento da área dos olhos e a análise ocular. Boa parte desses sistemas utilizam como base as métricas de *PERCLOS* (*Percentage of closure - Porcentagem de fechamento*) e *AVECLOS* (*Average of eye closure - Média de fechamento dos olhos*). O *PERCLOS* é definido como o percentual de tempo (em um minuto) que os olhos estão ao menos 80% fechados. Esse termo também é conhecido como *PERCLOS 80*. Há ainda aplicações que utilizam o *PERCLOS 70*, que se refere ao percentual de tempo em um minuto, em que os olhos estão pelo menos 70% fechados. Já a *AVECLOS* consiste no percentual de tempo em um minuto, em que os olhos estão totalmente fechados (MARSHALL;

PLEYDELL-PEARCE; DICKSON, 2003).

O NHTSA (*National Highway Traffic Safety Administration* - Administração Nacional de Segurança no Trânsito Rodoviário), que é o departamento de trânsito dos Estados Unidos, investigou a relação entre *PERCLOS* e outras medidas de atenção e identificou que existe uma alta correlação entre a diminuição da atenção e o *PERCLOS* 80 (DINGES D.F.; MALLIS, 1998). Além disso, em um estudo realizado por HANOWSKI (2008), foi avaliado o estado de sonolência do motorista considerando o valor do *PERCLOS* 80 a cada três minutos. Neste caso, foi observado que o motorista encontrava-se atento quando o valor do *PERCLOS* 80 estava entre 0% e 9%, em um estado suspeito com um valor entre 9% e 12%, e com uma baixa atenção quando o valor era maior que 12%. Por fim, no trabalho desenvolvido por BERGASA *et al.* (2006) foi utilizado o valor do *PERCLOS* 80 para a identificação da sonolência, obtendo-se uma taxa de sucesso próxima a 100%. No entanto, essa alta taxa de acerto foi obtida quando o motorista não usava óculos.

Como exemplos de sistemas que utilizam essa técnica pode-se destacar: o *Co-Pilot*, que foi desenvolvido por Richard Grace do instituto de robótica da Universidade Carnegie Mellon, da Pensilvânia (GRACE; STEWARD, 2001); o *Optalert* que foi desenvolvido pela empresa *Sleep Diagnostics Ltd.*; o *Insight* que foi criado pela empresa *SMI*; o *Driver Fatigue Monitor* da empresa *Attention Technologies*; o *FaceLab* desenvolvido pela empresa *Seeing Machines*; e o *Driver State Monitor* que foi desenvolvido pela empresa *Delphi's Corporation* (EDWARDS *et al.*, 2007).

2.3.2 Postura da cabeça

A postura da cabeça do motorista também pode ser utilizada para identificar um aumento no nível de fadiga, uma vez que um motorista cansado tende a perder a rigidez muscular do pescoço. Portanto, a cabeça pode começar a ter uma posição ou movimento diferente, seja queda ou movimento para os lados, principalmente quando o motorista está cansado ou sonolento (DINGES *et al.*, 1998). Assim, os movimentos laterais com a cabeça são um sinal comum de sonolência. Além disso, a posição da cabeça também pode ser utilizada para medir o estado de atenção do motorista. Por exemplo, se o motorista estiver olhando para o lado, esse possivelmente encontra-se desatento.

Em sistemas que utilizam a inclinação do pescoço para a identificação do estado de sonolência, basicamente é verificado o ângulo de inclinação da cabeça do motorista, sendo que, quando o ângulo ultrapassar um certo limite, um alarme sonoro é emitido (SAINI; SAINI, 2014). Esse tipo de técnica é comumente utilizada em conjunto com outras técnicas. A empresa *Smart Eye*, por exemplo, criou um sistema para a identificação de sonolência que utiliza uma luz infravermelha e uma câmera para verificar a posição da cabeça e os atributos dos olhos (CHACON-MURGUÍA; PRIETO-RESENDIZ, 2015).

2.3.3 Identificação do Bocejo

Juntamente com a análise dos olhos e da postura da cabeça, a análise da boca é um dos sinais físicos mais relevantes para a detecção de sonolência, já que o bocejo é uma atividade colateral habitualmente causada pela sonolência. A detecção da sonolência com base no bocejo envolve etapas que comumente incluem, a detecção em tempo real e o rastreamento facial (etapa em comum com a análise da posição da cabeça e também com a análise dos olhos), para então executar a detecção e o rastreamento do contorno da boca. Assim, a partir de mudanças no contorno e na abertura da boca, é possível determinar se o motorista está bocejando ou não (SAINI; SAINI, 2014). Segundo SAINI; SAINI (2014), é importante que seja realizada primeiramente uma detecção da face e depois a detecção da boca, a fim de evitar erros na detecção.

Um exemplo da utilização dessa abordagem para a identificação do bocejo pode ser encontrado no sistema desenvolvido por YU *et al.* (2019). Neste, o estado do motorista não é definido utilizando apenas uma imagem, já que estar com a boca aberta pode simplesmente significar que o motorista esteja falando. Assim, para a detecção da sonolência é utilizada uma janela de detecção com comprimento de 20 quadros, sendo que um bocejo é detectado, quando o motorista estiver com a boca aberta em ao menos 15 quadros (3/4 dos quadros).

3 DETECÇÃO DO ESTADO DE SONOLÊNCIA ATRAVÉS DO PROCESSAMENTO DE IMAGENS

Os sistemas baseados no processamento de imagens para a detecção de sonolência são sistemas não intrusivos, uma vez que não necessitam de nenhum tipo de sensor acoplado ao motorista. Esses são mais recomendados, pois são mais confortáveis e não interferem no motorista garantindo uma condução mais segura. Esses sistemas geralmente envolvem as etapas de: captura e pré-processamento das imagens, métodos de detecção e extração facial, detecção e rastreamento dos olhos e da boca e, por fim, a definição do estado de sonolência. Neste capítulo, é realizado um levantamento bibliográfico sobre as principais técnicas utilizadas em cada uma dessas etapas. Além disso, são apresentados trabalhos relacionados, que contribuíram de alguma forma para o desenvolvimento deste trabalho.

3.1 CAPTURA E PRÉ-PROCESSAMENTO DAS IMAGENS

O processo de captura das imagens geralmente é realizado utilizando uma câmera acoplada ao painel do veículo, onde o que varia é o tipo de dispositivo utilizado para a captura das imagens. Por exemplo, nos trabalhos desenvolvidos por ALSHAQAQI *et al.* (2013) e TIPPRASERT *et al.* (2019), o processo de captura de imagens é realizado utilizando uma câmera de infravermelho. Já no trabalho desenvolvido por INTHANON; MUNGSING (2020) esse procedimento é realizado através de uma *webcam*. Por fim, nos trabalhos desenvolvidos por YU *et al.* (2019) e DASGUPTA; RAHMAN; ROUTRAY (2018) é utilizada a câmera de um *smartphone*.

Após a captura das imagens, alguns sistemas realizam a detecção facial diretamente, ou seja, nenhum pré-processamento é realizado sobre as imagens. Como exemplo, pode-se citar os trabalhos desenvolvidos por TIPPRASERT *et al.* (2019), YU *et al.* (2019), ALIOUA; AMINE; RZIZA (2014) e INTHANON; MUNGSING (2020). No entanto, essa abordagem pode afetar o desempenho do sistema, devido, principalmente, a existência de ruídos nas imagens, baixo contraste das imagens, imagens de rosto no banco traseiro, etc. Desta forma, é comum a utilização de etapas de pré-processamento das imagens, que possuem como principal objetivo otimizar as etapas subsequentes, seja em ganho de performance ou diminuição dos erros de detecção (DASGUPTA; RAHMAN; ROUTRAY, 2018).

Na etapa de pré-processamento, geralmente são utilizadas etapas relacionadas com o corte das imagens, conversão para tons de cinza, o aumento no contraste das imagens, filtros de detecção de bordas, correção geométrica, redimensionamento da imagem, etc. As etapas de pré-processamento a serem utilizadas variam de acordo com a abordagem utilizada para a detecção da região de interesse, seja ela a face, olhos e boca (ALSHAQAQI *et al.*, 2013; DASGUPTA; RAHMAN; ROUTRAY, 2018).

A conversão para escala cinza é utilizada geralmente para reduzir o custo computacional, sendo utilizada em casos onde a informação da cor da imagem é irrelevante para as análises posteriores. Essa conversão é comumente utilizada em aplicações que empregam um Classificador *Haar*, por exemplo, para a detecção da face (VIOLA; JONES, 2001). Esse método compara a diferença de intensidade de níveis de cinza em diferentes regiões da imagem, necessitando da imagem em tons de escala cinza. De fato, essa conversão é realizada nos trabalhos desenvolvidos por (DASGUPTA; RAHMAN; ROUFRAY, 2018; TIPPRASERT *et al.*, 2019), que utilizam um Classificador *Haar*.

O aumento de contraste da imagem é usualmente utilizado para aumentar o contraste global da imagem, principalmente para evidenciar a diferença entre a região da face e o fundo da imagem. Com isso evita-se erros na detecção em casos onde a intensidade de *pixels* da imagem esteja concentrada em uma faixa específica, seja ela muito escura ou muito clara. Para tanto, os métodos mais utilizados baseiam-se na equalização do histograma, onde as frequências de distribuição de intensidades dos *pixels* são distribuídas uniformemente. Neste caso, também é comum a utilização de técnicas de equalização com uma limitação do contraste, de forma a evitar que o aumento do contraste gere áreas saturadas e com muito brilho. Como exemplo, pode-se citar o trabalho desenvolvido por (DASGUPTA; RAHMAN; ROUFRAY, 2018), onde foi utilizado o *CLAHE* (*Contrast Limited Adaptive Histogram Equalization*), que consiste em uma técnica de equalização de histograma com limitação de contraste (ver Figura 1).



Figura 1 – Equalização de histograma com limitação de contraste

Fonte: (DASGUPTA; RAHMAN; ROUFRAY, 2018)

Os filtros para a detecção de bordas são frequentemente utilizados em sistemas que realizam a identificação facial através do contorno da face. No trabalho desenvolvido por KATIYAR; ARUN (2014) é feita a comparação entre os principais filtros de detecção de bordas, como por exemplo, os filtros de *Canny*, *Prewitt*, *Sobel*, *Roberts*, *Laplacian* e *Zero crossing*. Neste estudo, concluiu-se que os dois melhores filtros são o *Canny* e *Sobel*, sendo que o filtro de *Sobel* apresenta um custo computacional menor que filtro de *Canny*, no entanto, em alguns casos provoca o surgimento de bordas falsas. De fato, os filtros de *Canny* e *Sobel* são os filtros mais utilizados em sistemas de detecção de sonolência. Por exemplo, no trabalho desenvolvido por ALSHAQAQI

et al. (2013), a identificação da face é realizada através da identificação da região em torno do ponto máximo do sinal de simetria. Neste caso, é utilizado um filtro de *Sobel* para realçar as bordas da face do motorista (ver Figura 2).



Figura 2 – Filtro de Sobel

Fonte: (ALSHAQAQI *et al.*, 2013)

A correção geométrica é utilizada quando o método de detecção facial adotado é sensível a posição da face sobre o plano da imagem, que geralmente tem que estar perpendicular a um ângulo de 90° . Esse método então é usado para rotacionar a imagem, caso a face não seja detectada no *frame* atual. Neste caso, o *frame* seguinte é rotacionado utilizando um determinado ângulo. Caso a face ainda não seja detectada, os *frames* posteriores também são rotacionados. Após a detecção da face, todos os *frames* seguintes são rotacionados utilizando esse ângulo (DASGUPTA; RAHMAN; ROUSTRAY, 2018). Como pode ser observado na Figura 3, no sistema desenvolvido por DASGUPTA; RAHMAN; ROUSTRAY (2018) essa técnica é utilizada, considerando uma rotação com uma variação no ângulo entre $[-60; -45; 0; 45; 60]$. Esta é uma etapa pouco utilizada, tendo em vista que em sistemas que analisam a sonolência do motorista, o ângulo da cabeça apresenta uma baixa variação que não impacta no reconhecimento da face.

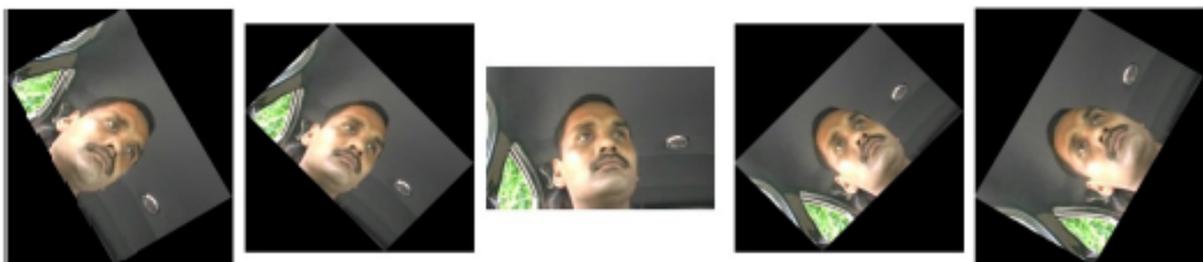


Figura 3 – Correção geométrica

Fonte: (DASGUPTA; RAHMAN; ROUSTRAY, 2018)

Outra técnica que é comumente utilizada é o redimensionamento da imagem, objetivando reduzir o tamanho da imagem e, conseqüentemente, o custo de processamento nas etapas posteriores. Essa etapa é muito utilizada em sistemas que necessitam de desempenho em tempo real e que executam em dispositivos com limitações de processamento. Ela consegue otimizar bastante os tempos de processamento do sistema, principalmente se é utilizada alguma técnica que aplica convolução, já que a convolução tem um alto custo computacional, e que aumenta significativamente conforme as dimensões da imagem analisada. As técnicas de redimensionamento de imagens mais utilizadas são a interpolação do vizinho mais próximo, interpolação bilinear e interpolação bicúbica. Por exemplo, no sistema desenvolvido por YU *et al.* (2019), a rede neural convolucional de arquitetura *MobileNet* utilizada, é alimentada com imagens de face, mas antes é feito redimensionamento para uma dimensão fixa de 300×300 pixels.

Neste trabalho são utilizadas duas etapas de pré-processamento, que são o redimensionamento da imagem e a conversão da mesma para tons de cinza. O redimensionamento da imagem torna-se necessário visto que o sistema é executado em um *smartphone* que possui limitações de capacidade de processamento. A conversão para tons de cinza é necessária uma vez que é utilizado um Classificador de *Haar* para a detecção da face, como será visto mais adiante. O procedimento de captura das imagens e as etapas de pré-processamento são descritas com maiores detalhes na Seção 4.1.

3.2 DETECÇÃO DA FACE

Existem diversos métodos que podem ser utilizados na etapa de detecção da face. Por exemplo, no trabalho desenvolvido por YU *et al.* (2019), é utilizada uma rede neural do tipo *MobileNet* para a detecção da face. Esta é uma arquitetura baseada em uma rede neural convolucional, criada com uma arquitetura simplificada e que utiliza a convolução separada em profundidade. A arquitetura *MobileNet* apresenta uma precisão muito próxima das redes neurais convolucionais tradicionais (HOWARD *et al.*, 2017), porém é mais eficiente e adequada para ser utilizada em dispositivos onde o poder de processamento é limitado, como por exemplo, dispositivos móveis.

No trabalho desenvolvido por ALIOUA; AMINE; RZIZA (2014) a detecção da face é feita através da utilização de Máquinas de Vetores de Suporte (*SVM - Support Vector Machines*). Durante a fase de treinamento, são utilizadas imagens de face e anti-face para o ajuste dos vetores de suporte. Então, é considerada uma janela de observação em todas as posições, escalas e orientações possíveis dentro da imagem. Cada vetor de suporte é pontuado por uma função não linear em relação à janela de observação, onde se a soma resultante estiver acima do limite pré estabelecido, a face é detectada.

No trabalho de ALSHAQAQI *et al.* (2013), para a identificação da face é obtido o sinal de simetria da imagem. Para a obtenção do sinal de simetria de uma imagem, isola-se os pixels

das bordas, utilizando um filtro de detecção de bordas. Os *pixels* das bordas são chamados de brancos e os demais *pixels* de pretos. Após, utiliza-se um vetor acumulador com tamanho igual à largura da imagem, sendo que para cada dois *pixels* brancos que estão na mesma linha da imagem, incrementa-se o valor correspondente à posição média entre esses dois *pixels* no vetor acumulador. A região da face é extraída da região em volta do ponto máximo do vetor acumulador. Na Figura 4 tem-se um exemplo que apresenta a identificação da face através do uso do sinal de simetria da imagem.

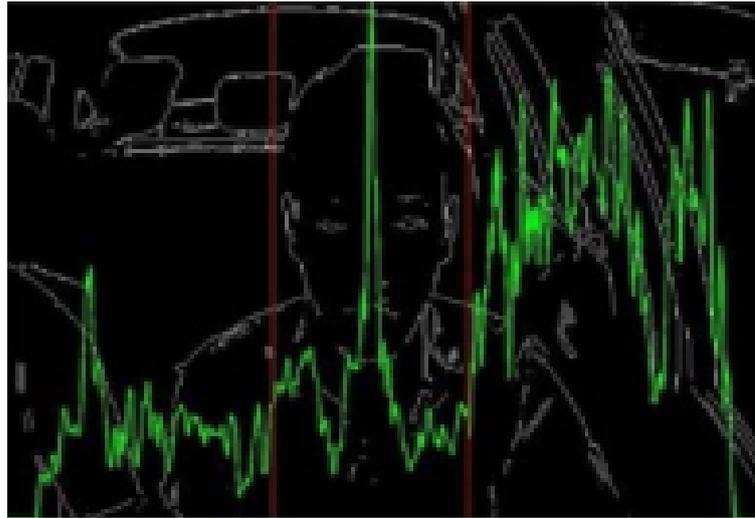


Figura 4 – Região de máxima do sinal de simetria

Fonte: (ALSHAQAQI *et al.*, 2013)

No entanto, o método mais utilizado para a detecção da face em sistemas de detecção de sonolência é o Classificador *Haar*. De fato, esse foi utilizado de forma satisfatória nos trabalhos desenvolvidos por DASGUPTA; RAHMAN; ROUSTRAY (2018), TIPPRASERT *et al.* (2019) e INTHANON; MUNGSING (2020). O Classificador *Haar* é um método de aprendizado de máquina que foi desenvolvido especificamente para a detecção facial. Ele utiliza-se de uma janela de detecção, similar a um *kernel* de convolução, que é movida sobre a imagem, comparando a diferença de intensidade dos *pixels* em cada uma das regiões. Para o desenvolvimento deste trabalho, foi utilizado um Classificador *Haar*, que será descrito com maiores detalhes na Seção 4.2.

3.3 DETECÇÃO DOS OLHOS E DA BOCA

Muitos dos métodos utilizados para a detecção da face podem ser utilizados também para a identificação da região de interesse, seja ela os olhos ou a boca. Por exemplo, no trabalho desenvolvido por TIPPRASERT *et al.* (2019), um Classificador *Haar* foi utilizado para a identificação da face e, posteriormente, outros dois Classificadores *Haar* foram utilizados para a identificação dos olhos e da boca. Após a identificação da face, a imagem é cortada pela metade, sendo que a parte superior é utilizada como entrada para um primeiro Classificador *Haar*,

que foi treinado com uma base de imagens com olhos. A parte inferior é utilizada como entrada para um segundo classificador que foi treinado utilizando um conjunto de imagens de bocas.

No trabalho desenvolvido por ALSHAQAQI *et al.* (2013), o reconhecimento dos olhos é realizado através do máximo sinal de simetria. Neste, após a identificação da face, é considerada somente a metade superior, uma vez que essa é a região onde se encontram os olhos. Então, varre-se verticalmente essa região utilizando-se uma máscara com uma altura estimada igual à altura dos olhos e uma largura igual à largura da face, calculando-se a simetria. A região dos olhos corresponde ao valor máximo do sinal de simetria. Após isso, essa nova região é particionada considerando a metade da esquerda e da direita. Para cada uma das partes é calculado o valor de máxima do sinal de simetria, que corresponde ao centro dos olhos (Figura 5).

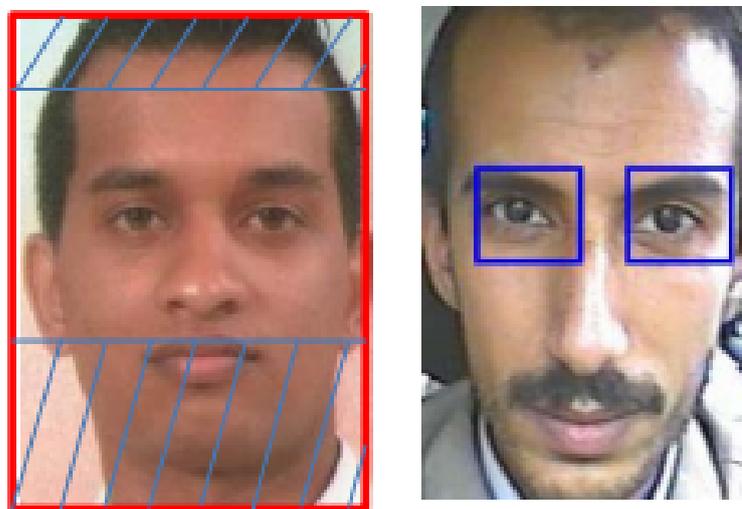


Figura 5 – Imagem dos olhos extraída ao redor da região de máxima do sinal de simetria

Fonte: (ALSHAQAQI *et al.*, 2013)

No trabalho desenvolvido por INTHANON; MUNGSING (2020), para a identificação dos olhos e da boca é utilizado o método de Kazemi (KAZEMI; SULLIVAN, 2014). Neste método, a intensidade de cada *pixel* da imagem é indexada através de um vetor, sendo utilizado um conjunto de árvores de regressão para a localização de pontos de referência faciais. A partir da utilização deste método, é possível identificar 68 pontos de referência facial.

Por fim, no trabalho desenvolvido por YU *et al.* (2019) foram utilizadas duas redes convolucionais do tipo *MobileNet*. A primeira delas foi treinada de forma a detectar a região da face. A segunda delas foi treinada de forma a identificar os olhos, boca, nariz e o contorno do rosto. Neste trabalho, optou-se pela utilização de uma rede neural convolucional chamada *BlazeFace*, que foi desenvolvida pela Google e utiliza uma arquitetura baseada na *MobileNet* para a classificação do *PERCLOS*. Essa permite extrair os pontos de referência facial e algumas classificações como, por exemplo, a probabilidade dos olhos estarem abertos ou fechados. Optou-se pela utilização de uma rede *BlazeFace*, pois essa arquitetura possui um bom desempenho, uma vez que foi desenvolvida especificamente para ser utilizada em aplicações móveis e embarcadas

(HOWARD *et al.*, 2017). A arquitetura de uma rede *BlazeFace* é descrita com mais detalhes na Seção 4.3.

3.4 DETECÇÃO DO ESTADO DE SONOLÊNCIA DO MOTORISTA

No trabalho de ALSHAQAQI *et al.* (2013), o estado de sonolência do motorista é determinado por *AVECLOS*. Neste a detecção da face, olhos e boca realizada é utilizando a máxima do sinal de simetria. Após a definição da região dos olhos é utilizada uma Transformada *Hough* para a identificação das formas circulares dos olhos. Caso seja identificada a existência de um círculo é considerado que o olho do motorista encontra-se aberto e, em caso contrário, é considerado que o olho encontra-se fechado. O motorista é considerado sonolento caso tenha os olhos fechados em pelo menos 5 *frames* sucessivos. A transformada circular de *Hough* é uma técnica que realiza a detecção de círculos em imagens através da criação de uma matriz acumuladora, onde cada *pixel* da imagem é representado por uma coordenada da matriz acumuladora. Para cada borda detectada na imagem original (usando o algoritmo Canny), é desenhado um círculo de raio r pré-determinado e são incrementados os valores da matriz acumuladora correspondentes as posições aonde o círculo passa. O ponto máximo da matriz acumuladora dá a localização do centro do círculo, como pode ser observado na Figura 6.

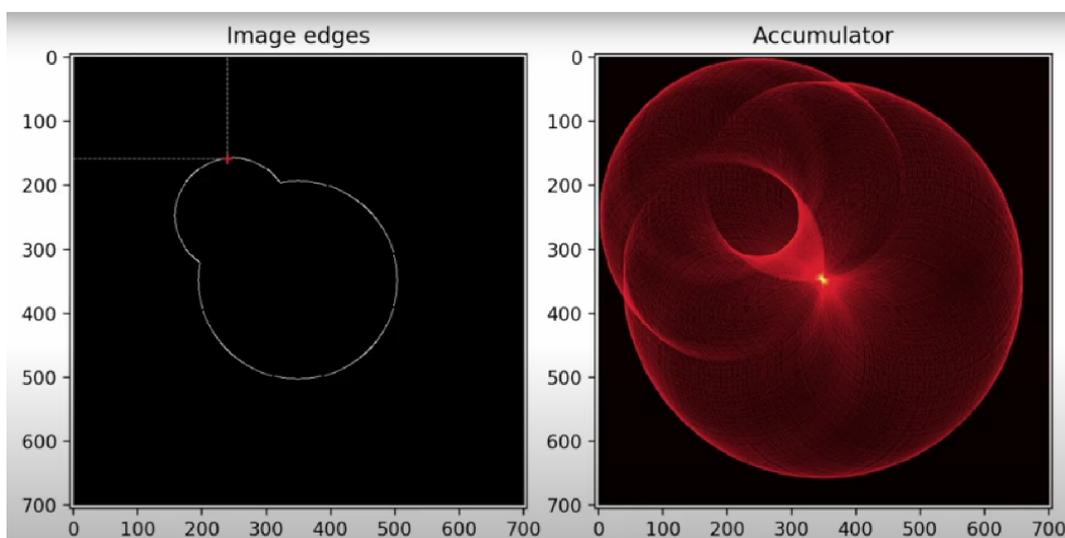


Figura 6 – Transformada de *Hough*

Fonte: (BRADSKI; KAEHLER, 2008)

No trabalho de TIPPRASERT *et al.* (2019), a face, olhos e boca são detectados utilizando um *Classificador Haar*. Após, considerando-se somente a região dos olhos, é utilizado um classificador *SVM* para a classificação das imagens, em imagens com os olhos abertos e com os olhos fechados. Um outro classificador *SVM* é utilizado na região da boca, classificando as imagens em, imagens com a boca aberta e com a boca fechada. A sonolência é identificada caso qualquer uma das duas áreas esteja com o sintoma de sonolência, ou seja, boca aberta em sinal de bocejo ou os olhos fechados.

No trabalho desenvolvido por INTANON; MUNGSING (2020), os pontos de referência dos olhos e da boca foram extraídos utilizando uma rede do tipo *MobileNet*. Com os pontos de referência isolados, a sonolência é definida com base em duas medidas. No caso dos olhos, a sonolência é detectada caso a distância entre os pontos de referência superiores e inferiores sejam próximos de 0. Neste caso, considerou-se que distâncias inferiores a 0,2 cm representam os olhos fechados. Para a boca, considerou-se a distância entre os pontos de referência localizados na parte superior e inferior da boca. Caso, essa distância fosse superior a 5 cm por mais de 2 segundos, foi considerado um provável bocejo.

No trabalho de ALIOUA; AMINE; RZIZA (2014), foi utilizado somente a análise da boca para a identificação do estado de sonolência. Mais especificamente, foi utilizada uma transformada circular de *Hough* para a identificação da forma circular da boca aberta. Neste caso, foi considerado um bocejo, caso fosse identificado, que a boca está aberta por mais de 2 segundos.

Por fim, no trabalho desenvolvido por DASGUPTA; RAHMAN; ROUSTRAY (2018), a identificação de olhos abertos ou fechados, foi realizada através de um classificador *SVM*. Esse foi treinado com 1.200 imagens de olhos (700 olhos abertos e 500 olhos fechados). Após a classificação das imagens entre olhos abertos ou fechados, a sonolência é identificada observando uma janela de tempo de 10 segundos. Neste caso, se o motorista apresentar os olhos fechados por mais de 2 segundos é considerado que esse apresenta sinais de sonolência.

Neste trabalho, a detecção do estado de sonolência foi realizada observando a probabilidade dos olhos estarem abertos ou fechados. Para o cálculo dessa probabilidade foi utilizado um classificador baseado nas arquiteturas *MobileNet* e *SSD*, chamado *BlazeFace*. O processo de detecção de sonolência é descrito mais detalhadamente na Seção 4.4.

4 DESCRIÇÃO DO SISTEMA DESENVOLVIDO

O sistema desenvolvido neste trabalho baseia-se no processamento de imagens para a detecção de sonolência. A detecção é realizada através da análise das características oculares do motorista. Na Figura 7 tem-se um fluxograma com as etapas que são utilizadas neste trabalho. Essas etapas foram definidas com base no trabalho desenvolvido por DASGUPTA; RAHMAN; ROUTRAY (2018), onde também é utilizado um *smartphone* para a captura das imagens. Nele também é realizada uma análise dos olhos para a detecção do estado de sonolência.

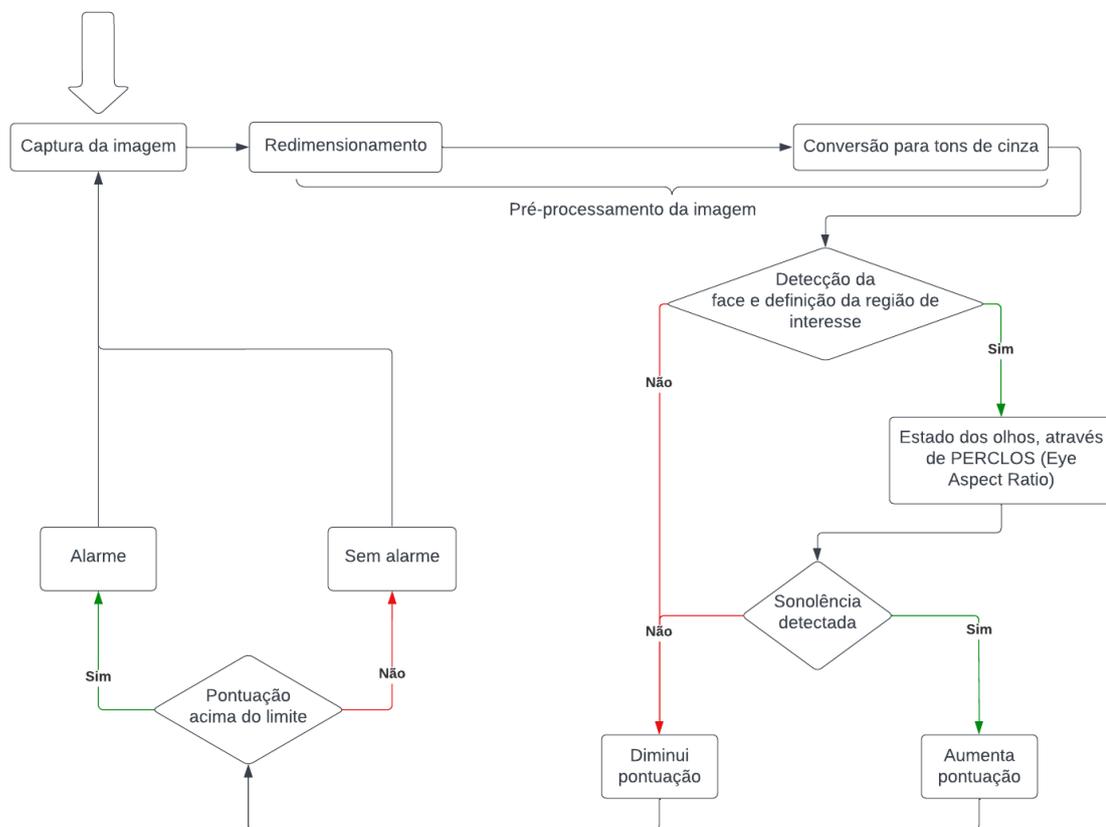


Figura 7 – Fluxograma com as etapas da proposta de solução

Fonte: O autor, 2022

A etapa da captura de imagens é essencial para o funcionamento adequado do sistema, visto que toda análise subsequente é dependente da qualidade da imagem capturada. Após a aquisição da imagem, esta é submetida a um processo de redimensionamento, reduzindo assim o número de *pixels* da imagem. Posteriormente, a imagem é convertida para tons de cinza para então ser aplicado o método de identificação da face. Por fim, a imagem da face é utilizada como entrada para classificar o percentual de abertura dos olhos (*PERCLOS*). A partir da identificação desse, combinado com os resultados de frames anteriores, é realizada a verificação se o motorista encontra-se em estado de sonolência.

4.1 CAPTURA E PRÉ-PROCESSAMENTO DAS IMAGENS

As imagens são capturadas em formato de vídeo utilizando um *smartphone*, que se encontra fixado no painel do carro, como pode ser observado na Figura 8. O celular utilizado é um *Galaxy M20* da *Samsung*. Esse possui uma câmera principal com uma resolução de 13 *megapixels* e abertura de F1.9 (SAMSUNG, 2019) e uma câmera frontal com resolução de 5 *megapixels*. Os vídeos são capturados utilizando uma resolução de 1920×1080 *pixels*, com uma taxa de atualização de 30 *frames* por segundo.



Figura 8 – Posição do *smartphone* para captura da imagem facial

Fonte: O autor, 2022

O pré-processamento das imagens tem início com um redimensionamento das imagens, objetivando reduzir o custo computacional das etapas posteriores. A partir do redimensionamento é possível reduzir a área total da imagem em *pixels*. De fato, a imagem de entrada, contém dimensões muito grandes, o que impacta consideravelmente no desempenho do sistema. O redimensionamento da imagem é feito através da técnica de interpolação do vizinho mais próximo. Essa técnica consiste em definir o *pixel* da imagem de saída através de uma média dos n *pixels* vizinhos da imagem de entrada. Para a solução proposta é feito um redimensionamento da imagem considerando uma largura fixa de 256 *pixels*. A altura é calculada de forma a manter a proporção da imagem de entrada. Testes foram realizados e neles as dimensões da imagem de entrada foram de 2448×3264 , então, cada *pixel* na imagem de saída é gerado pela média de 9 *pixels* da imagem de entrada. O redimensionamento das imagens não reduziu a precisão das etapas de detecção facial e identificação da abertura dos olhos, e provocou um expressivo ganho de performance.

Após o redimensionamento, as imagens são convertidas para uma escala de tons de cinza, uma vez que as informações de cor não são utilizadas pelo Classificador Haar para a identificação da face (BRADSKI; KAEHLER, 2008). Além disso, a conversão para tons de cinza possibilita uma redução no tamanho das imagens. Para a conversão da imagem para a escala de tons de cinza, o valor de cada *pixel* é obtido através da média ponderada das componentes RGB, como pode ser observado na Equação 4.1,

$$p = 0.299 * R + 0.587 * G + 0.114 * B \quad (4.1)$$

onde R , G e B , correspondem aos coeficientes das componentes, vermelho (*red*), verde (*green*) e azul (*blue*), respectivamente. Na Figura 9 tem-se um exemplo da imagem capturada convertida para tons de cinza.



Figura 9 – Imagem após conversão para escala cinza

Fonte: O autor, 2022

4.2 DETECÇÃO DA FACE

Para detecção da face foi utilizado o Classificador *Haar*, uma vez que esse método apresenta um bom desempenho e uma baixa taxa de erro (DASGUPTA; RAHMAN; ROUTHAY, 2018; TIPPRASERT *et al.*, 2019; INTHANON; MUNGSING, 2020). Este é um método de detecção de objetos que foi proposto por *Paul Viola* e *Michael Jones* em 2001, e que se baseia em uma abordagem de aprendizado de máquina para a detecção de objetos, e que foi desenvolvido especificamente para a detecção facial (VIOLA; JONES, 2001).

O classificador *Haar* é um método que se baseia em 4 etapas, que são: cálculo dos recursos de Haar; criação das imagens integrais; realização de uma etapa de treinamento utilizando um algoritmo de aprendizado baseado no método *AdaBoost* (*Adaptive Boosting*); e a utilização de classificadores em cascata.

Na primeira etapa são calculados os recursos de *Haar*, que são utilizados, posteriormente, para o reconhecimento dos objetos. O cálculo dos recursos *Haar* envolve somar as intensidades dos *pixels* em diferentes regiões (retângulos) da imagem e calcular a diferença entre os somatórios dessas regiões. A diferença de intensidade dos *pixels* (recursos de *Haar*) entre as regiões, é uma característica que pode ser utilizada para detecção da face, uma vez que o rosto humano compartilha de algumas propriedades, como por exemplo, a região dos olhos é mais escura que as regiões acima (testa e o nariz) e a região do nariz é mais brilhante que as regiões ao lado (olhos e bochecha). As características não são aplicadas na imagem como um todo. Elas são processadas e aplicadas movendo-se uma janela de detecção sobre a imagem de entrada, como um *kernel* de convolução (Figura 10).

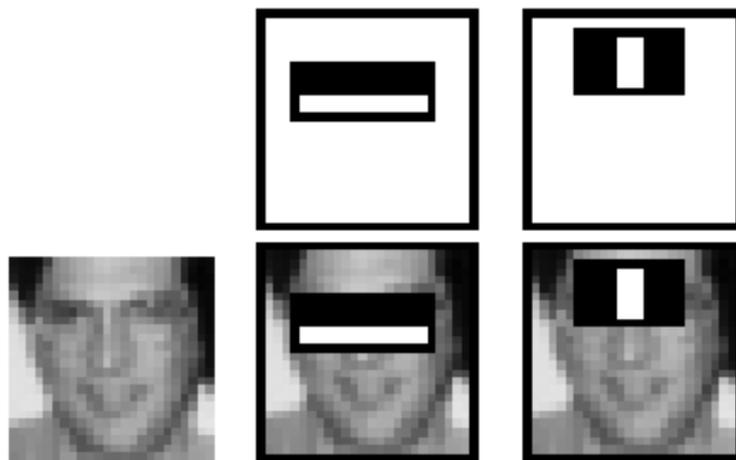


Figura 10 – Recursos de *Haar* selecionados após a etapa de treinamento

Fonte: (VIOLA; JONES, 2001)

Como pode ser observado na Figura 11, existem basicamente três tipos de recursos de *Haar*: os recursos de dois retângulos, os de três retângulos e os de quatro retângulos. Os recursos de dois retângulos são calculados através da diferença entre a soma dos *pixels* dentro de duas regiões retangulares. O valor de um recurso de três retângulos é obtido através da soma de dois retângulos externos subtraídos da soma em um retângulo central. Já o recurso de quatro retângulos é calculado através da diferença entre os pares diagonais de retângulos.

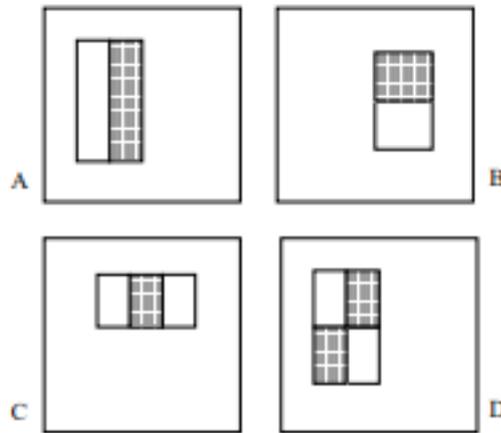


Figura 11 – Recursos de *Haar*: a soma dos *pixels* dos retângulos brancos é subtraída da soma dos *pixels* dos retângulos cinza

Fonte: (VIOLA; JONES, 2001)

O cálculo dos recursos *Haar* em imagens grandes apresenta um elevado custo computacional. Assim, para reduzir o número de operações, utiliza-se o conceito de imagem integral. A imagem integral, que também é conhecida como tabela de soma de áreas, foi proposta por *Frank Crow* em 1984 e permite calcular de forma mais eficiente a soma das intensidades dos *pixels* de uma área retangular.

A imagem integral no ponto $[x, y]$ contém a soma das intensidades de todos os *pixels* acima e à esquerda do ponto x e y . Por exemplo, na Figura 12 a região foi dividida nos sub-retângulos *A*, *B*, *C* e *D* com os pontos de referência 1, 2, 3 e 4, respectivamente. Com isso, tem-se que o valor da imagem integral no ponto 1 é dada pela soma das intensidades dos *pixels* do sub-retângulo *A*. Já o valor no ponto 2 é dado pela soma das intensidades dos *pixels* dos sub-retângulos *A* e *B* ($A + B$). No ponto 3 o valor da imagem integral é dado pela soma das intensidades dos *pixels* do sub-retângulos *A* e *C* ($A + C$). Por fim, no ponto 4 a imagem integral é dada pela soma das intensidades dos *pixels* dos sub-retângulos *A*, *B*, *C* e *D* ($A + B + C + D$).

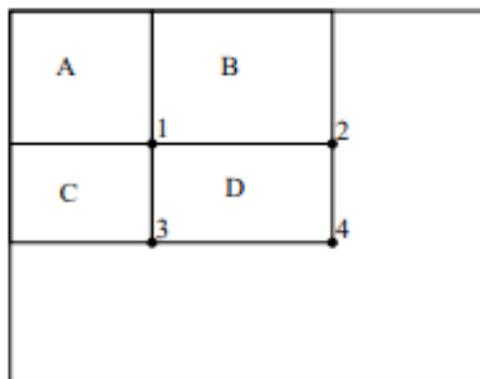


Figura 12 – Exemplo de aplicação da imagem integral

Fonte: (VIOLA; JONES, 2001)

Os valores da imagem integral são computados para cada *pixel* em uma única varredura na imagem, onde cada *pixel* é armazenado em uma matriz, com o seu valor de intensidade somado com os valores dos *pixels* anteriores. Com isso, obtêm-se uma matriz do tamanho da imagem, onde cada elemento da matriz, possui a soma de todos os níveis de cinza dos *pixels* à esquerda e acima do *pixel* atual (ver Figura 13). A imagem integral é utilizada posteriormente para o cálculo de qualquer um dos recursos *Haar*. Por exemplo, para calcular a intensidade dos *pixels* do retângulo de 3×3 destacado na Figura 13 (a), são necessárias 9 operações de soma ($20+50+20+50+100+50+20+50+20 = 380$) e esse custo aumenta conforme as dimensões do retângulo também aumenta. Já, para calcular a intensidade dos *pixels* desse mesmo retângulo utilizando a imagem integral da Figura 13 (b), são necessárias apenas 4 operações de soma ($(1 + 398) - (9 + 10) = 380$), sendo que esse custo independe das dimensões do retângulo.

1	2	5	1	2
2	20	50	20	5
5	50	100	50	2
2	20	50	20	1
1	5	25	1	2
5	2	25	2	5
2	1	5	2	1

1	3	8	9	11
3	25	80	101	108
8	80	235	306	315
10	102	307	398	408
11	108	338	430	442
16	115	370	464	481
18	118	378	474	492

a) Imagem de entrada

b) Imagem integral

Figura 13 – Representação da imagem de entrada em um tamanho de 5×7 *pixels* com a intensidade de cada *pixel*, ao lado, sua imagem integral

Fonte: (BRADSKI; KAEHLER, 2008)

Posteriormente, é realizada uma etapa de treinamento utilizando um algoritmo de aprendizado de máquina baseado no método *AdaBoost*. Durante essa etapa, são selecionados os recursos *Haar* que classificam com maior precisão as imagens faciais e não faciais (VIOLA; JONES, 2001). Para cada recurso *Haar*, um classificador é treinado utilizando apenas esse recurso, sendo que cada classificador recebe um peso igual. Após, os classificadores que apresentaram um melhor desempenho, possuem seu peso aumentado conforme Equação 4.2, onde $e(i) = 0$ corresponde aos casos onde a imagem foi classificada corretamente e $e(i) = 1$ os casos onde não foi classificada corretamente. Já o coeficiente β é obtido pela equação $\frac{e(t)}{1-e(t)}$, onde $e(t)$ corresponde ao erro. Esse processo é repetido utilizando os pesos atualizados e as novas taxas de erros são calculadas. Novamente, os pesos dos classificadores são ajustados e a classificação é reexecutada. Esse processo é repetido até que um limiar definido seja alcançado. O classificador final é obtido pela combinação ponderada dos classificadores obtidos por cada um dos recursos de *Haar* (VIOLA; JONES, 2001).

$$\omega(t + 1, i) = \omega(t, i)\beta(t)^{1-e(i)} \quad (4.2)$$

Para um melhor desempenho no processo de detecção e uma redução no tempo de processamento, os classificadores obtidos na etapa de *Adaboost* são organizados em cascata, como pode ser observado na Figura 14. Posteriormente, ao varrer uma imagem de entrada pela janela de detecção, se uma região analisada pelo primeiro nível da cascata é aceita, essa região é direcionada para o próximo estágio, seguindo até o último estágio. Caso ela seja aceita em todos os estágios da cascata, é realizada a identificação da face. Caso essa região seja rejeitada em qualquer estágio, ela é imediatamente descartada. Dessa forma, as regiões que não possuem a face são imediatamente descartadas pelo classificador inicial, não sendo analisadas pelos outros classificadores.

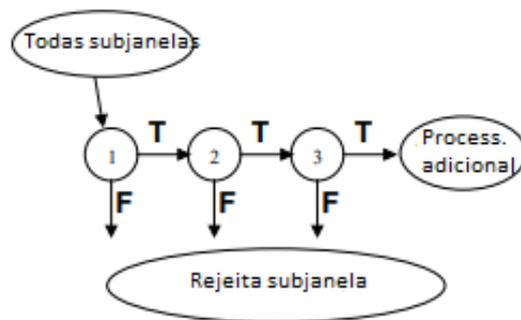
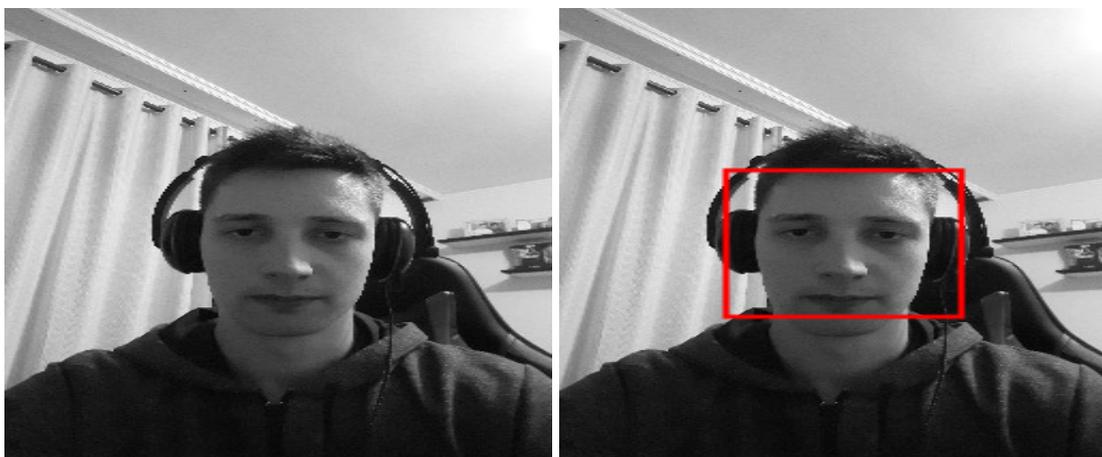


Figura 14 – Representação esquemática de uma cascata de detecção

Fonte: (VIOLA; JONES, 2001)

Nos testes realizados, para o treinamento, foram utilizadas 4916 faces rotuladas manualmente e 9544 imagens sem face que também foram inspecionadas de forma manual. Para o sistema proposto, foi definida uma região mínima de tamanho de 180×180 pixels para detecção do rosto. Desta forma, imagens de face pequenas, como a de uma pessoa no banco de trás do veículo, não são detectadas. Na Figura 15 tem-se um exemplo do resultado da aplicação do classificador de *Haar*.



a) Imagem após o pré-processamento

b) Imagem após face detectada

Figura 15 – Imagem após a aplicação do método *Classificador Haar*

Fonte: O autor, 2022

4.3 RECONHECIMENTO DOS OLHOS

A área em torno da face é recortada sendo utilizada para o cálculo do *PERCLOS*. Neste caso para a identificação dos olhos, optou-se pela utilização de uma rede convolucional do tipo *BlazeFace*. Essa arquitetura é uma rede de extração de recursos faciais que foi desenvolvida pela empresa Google. Ela se caracteriza por ser uma rede neural convolucional que não necessita de muitos recursos computacionais, sendo muito utilizada em dispositivos móveis e em sistemas embarcados, já que combina ótima precisão (98,61%) com alto desempenho (BAZAREVSKY *et al.*, 2019).

O desenvolvimento da arquitetura *BlazeFace* foi realizado baseado nas arquiteturas *MobileNet* e *SSD (Single Shot Multibox Detector)*. A arquitetura *MobileNet* é uma forma de construir modelos de redes neurais convolucionais, desenvolvida por pesquisadores da Google, que baseia-se em redes que utilizam convolução separada em profundidade (*depthwise separable convolution*). A convolução separada em profundidade é realizada a partir de duas operações, a primeira é a etapa de filtragem, onde é aplicada a convolução em profundidade (*depthwise convolution*) e a segunda é a etapa de combinação, onde é aplicada a convolução pontual (*pointwise convolution*) (HOWARD *et al.*, 2017).

A convolução é uma medida de sobreposição entre duas funções quando uma delas desliza sobre a outra. Em imagens, essa operação é efetuada movendo um *kernel* sobre imagem de entrada e somando o produto do *kernel* pelo valor da imagem de entrada em torno de cada um dos *pixel*. O problema deste método é seu elevado custo computacional, visto que os *kernels* são aplicados a todos os canais de entrada. Por exemplo, em uma camada de convolução padrão, considerando-se como entrada um mapa de características F de dimensões $D_f \times D_f \times M$ e N *kernels* de convolução de dimensões $D_k \times D_k \times M$ (Figura 16), tem-se que o custo computacional dessa operação de convolução é de $D_k^2 \times M \times N \times D_f^2$.

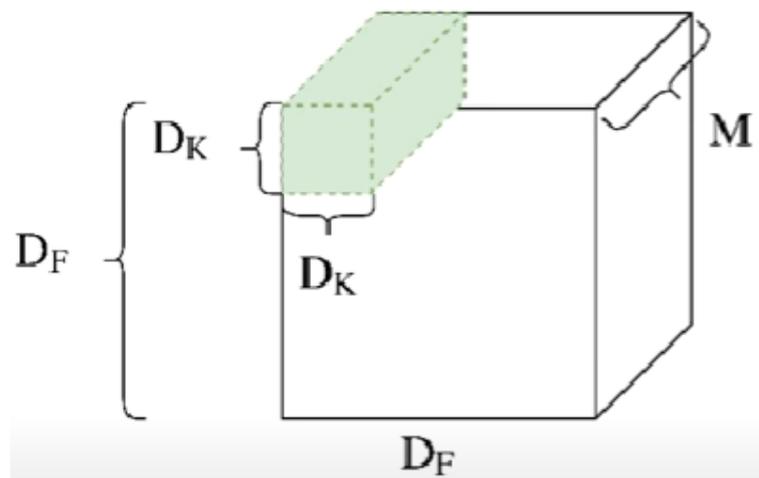


Figura 16 – Mapa de características F e kernel de convolução K

Fonte: (HOWARD *et al.*, 2017)

Já em operações de convolução em profundidade (*depthwise convolution*) os *kernels* são aplicados considerando-se um único canal de entrada. Por exemplo, no caso do mapa de características F de dimensões $Df \times Df \times M$ ao invés de utilizar *kernels* de tamanho $Dk \times Dk \times M$ utiliza-se *kernels* de tamanho $Dk \times Dk \times 1$. Desta forma, a convolução será aplicada apenas a um canal de entrada (Figura 17), apresentando um custo computacional de $Dk^2 \times Df^2$ para cada *kernel* e, conseqüentemente, quando aplicada para M *kernels* tem-se um custo computacional de $Dk^2 \times M \times Df^2$.

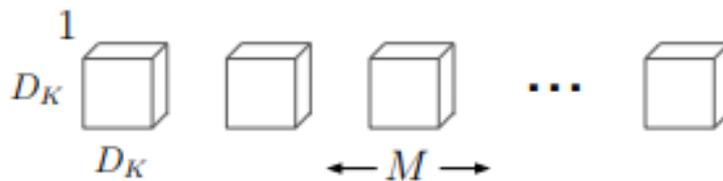


Figura 17 – *Kernels* de convolução em profundidade

Fonte: (HOWARD *et al.*, 2017)

No entanto, este método é utilizado apenas para filtrar o canal de entrada, não produzindo novos recursos. Desta forma, essa etapa é seguida por uma etapa de combinação. Nela, a saída da primeira etapa é combinada através de uma convolução ponto a ponto, na qual é realizada uma combinação linear da saída da convolução em profundidade usando uma convolução 1×1 . Por exemplo, utilizando-se o mapa de recursos F de dimensões $Df \times Df \times M$ e utilizando-se *kernels* apenas com a dimensão de profundidade M ($1 \times 1 \times M$), como pode ser observado na Figura 18, tem-se um custo computacional de $M \times Df^2$ para cada *kernel*. Aplicando-se para N *kernels* tem-se um custo computacional total de $M \times Df^2 \times N$.

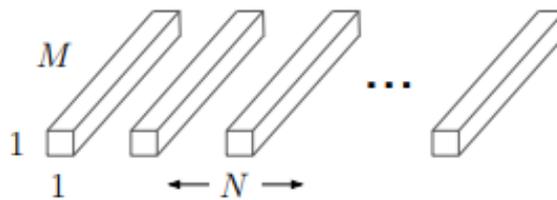


Figura 18 – Filtros convolucionais 1×1 chamados de filtros de convolução pontual (*Pointwise*) no contexto da convolução separável em profundidade

Fonte: (HOWARD *et al.*, 2017)

A partir da soma dos custos computacionais das duas etapas, divididos pelo custo computacional de uma convolução padrão, obtêm-se a Equação 4.3. Desta forma, considerando-se um *kernel* de convolução de dimensões 3×3 e uma imagem RGB de 1024 canais de saída, o custo computacional utilizando a convolução separada em profundidade será $\frac{1}{1024} + \frac{1}{9} = 0,112$ do tempo de execução de uma convolução padrão. Essa diferença no custo computacional aumenta conforme se utiliza mais canais de saída e *kernels* de dimensões maiores.

$$\frac{Dk^2 \cdot Df^2 \cdot M + M \cdot Df^2 \cdot N}{Dk^2 \cdot M \cdot N \cdot Df^2} = \frac{1}{N} + \frac{1}{Dk^2} \quad (4.3)$$

Como pode ser observado na Figura 19, cada uma das etapas de convolução é seguida de uma etapa de normalização de lote (*BN - Batch Normalization*) e de uma função de ativação do tipo unidade linear retificada (*ReLU - Rectified Linear Unit*). Além disso, para a construção de modelos ainda mais eficientes, são utilizados dois hiperparâmetros chamados de multiplicador de largura (α) e multiplicador de resolução (ρ). O multiplicador de largura α é um valor entre 0 e 1, que é utilizado para reduzir uma rede uniformemente em cada camada. Ou seja, para cada camada, o valor de α será multiplicado pelos canais de entrada e saída, M e N . Assim, tem-se uma redução do custo computacional para $Dk \times Dk \times \alpha M \times Df \times Df + \alpha M \times \alpha N \times Df \times Df$. Já o multiplicador de resolução ρ , também consiste em um valor entre 0 e 1, sendo aplicado ao mapa de recursos de entrada Df . Com isso, o custo computacional é reduzido para $Dk \times Dk \times M \times \rho Df \times \rho Df + M \times N \times \rho Df \times \rho Df$.

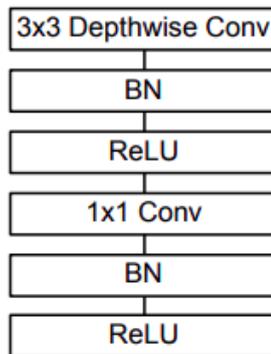


Figura 19 – Camadas da rede neural convolucional *MobileNet*

Fonte: (HOWARD *et al.*, 2017)

A redução do custo computacional impacta em uma redução de precisão, comparado a outros modelos de redes neurais convolucionais. No entanto, essa redução de precisão não é significativa e a rede apresenta um tempo de execução inferior. Na Tabela 1 tem-se uma comparação entre a precisão e o tempo de execução de uma rede neural convolucional do tipo *MobileNet* e de uma *VGG16*. Como pode ser observado, essas apresentam uma precisão muito similar, porém a rede *MobileNet* apresenta um tempo de execução quase 33 vezes menor que uma rede *VGG16*.

Tabela 1 – *MobileNet* vs outros modelos

Modelo	Precisão	Multiplicações e adições (milhões)	Parâmetros (milhões)
<i>MobileNet</i>	70,6%	569	4,2
<i>GoogLeNet</i>	69,8%	1550	6,8
<i>VGG16</i>	71,5%	15300	138

Fonte: (HOWARD *et al.*, 2017)

Em termos de custo computacional, o aumento do tamanho do *kernel* de convolução em profundidade é relativamente pequeno. Assim, considerando uma rede *MobileNet*, onde os cálculos de convolução são separados em convolução em profundidade e pontual, tem-se que o tempo de execução é dominado pela etapa de convolução pontual. De fato, em testes realizados por BAZAREVSKY *et al.* (2019), utilizando um *smartphone iPhone X*, verificou-se que o tempo necessário para a execução da etapa de convolução foi 0,07 ms para um mapa de características de dimensões $56 \times 56 \times 128$. Já o tempo de execução necessário para a execução da convolução pontual nos 128 canais apresentou um tempo de execução de 0,3 ms.

A partir dessa observação conclui-se que aumentar o tamanho do *kernel* da parte em profundidade é relativamente barato. Assim, em uma rede neural do tipo *BlazeFace*, tem-se um aumento das dimensões do *kernel* de convolução em profundidade de uma arquitetura *MobileNet* de 3×3 para 5×5 . Apesar de uma pequena perda de desempenho na etapa de convolução em profundidade tem-se um aumento de desempenho na etapa de convolução pontual, visto que com o aumento dos *kernels* na etapa de convolução em profundidade, tem-se um aumento na dimensão das camadas de saída permitindo uma diminuição do número de canais utilizados na convolução pontual.

Por exemplo, em um mapa de características de dimensões $56 \times 56 \times 128$ e um *kernel* de dimensões 3×3 , a etapa de convolução em profundidade gera sua camada de saída de dimensões $56 \times 56 \times 128 \times 3 \times 3$, resultando uma saída de 3.612.672 valores. Já, utilizando um *kernel* 5×5 , a etapa de convolução em profundidade gera uma camada de saída de dimensões $56 \times 56 \times 128 \times 5 \times 5$, resultando em uma saída de 10.035.200. Assim, a saída utilizando um *kernel* 3×3 tem um tamanho que corresponde a 36% do tamanho utilizando um *kernel* 5×5 . No entanto, uma saída de mesmo tamanho pode ser obtido utilizando um número de canais menor na etapa de convolução pontual, reduzindo o custo computacional da etapa de convolução pontual. (BAZAREVSKY *et al.*, 2019).

As redes neurais do tipo *BlazeFace*, se caracterizam ainda por utilizarem caixas delimitadoras com tamanho fixos, chamadas de âncoras. Essas caixas delimitadoras são utilizadas em modelos de detecção de objetos do tipo *SSD*. Essas caixas delimitadoras são utilizadas em cada camada da rede neural, onde é avaliada a presença do objeto a ser detectado em cada uma das caixas delimitadoras. Em uma aplicação da arquitetura *SSD*, o tamanho das caixas delimitadoras variam entre tamanhos de 1×1 , 2×2 , 4×4 , 8×8 e 16×16 . Em uma rede neural do tipo

BlazeFace são utilizadas caixas delimitadoras de tamanho 8×8 e 16×16 . O tamanho dessas caixas delimitadoras foram definidas considerando a variação das proporções do rosto humano (BAZAREVSKY *et al.*, 2019).

Uma rede *BlazeFace* possibilita a identificação de pontos de referência facial, que podem ser utilizados para determinar a posição dos olhos, boca, orelhas e nariz. Além disso, esse tipo de arquitetura retorna algumas classificações como, por exemplo, a probabilidade de existência de um sorriso e a probabilidade dos olhos direito e esquerdo estarem abertos. Esse modelo foi originalmente treinado por BAZAREVSKY *et al.* (2019), utilizando-se um conjunto de 66000 imagens de faces rotuladas com os 6 pontos de referência facial. Na Figura 20, tem-se um exemplo dos resultados obtidos utilizando uma rede do tipo *BlazeFace*, onde é possível visualizar 6 pontos de referência facial (destacados em azul).

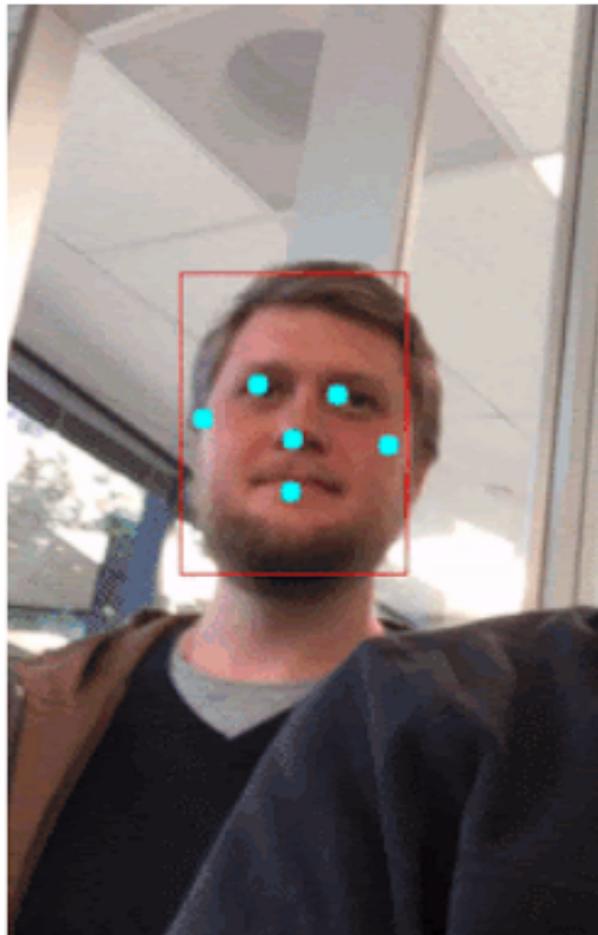


Figura 20 – Imagem do rosto após a aplicação do método de *BlazeFace*. Os 6 pontos de referência facial estão destacados em azul

Fonte: (BAZAREVSKY *et al.*, 2019)

4.4 DETECÇÃO DO ESTADO DE SONOLÊNCIA

A detecção de sonolência é realizada considerando as probabilidades dos olhos direito e esquerdo estarem abertos. Essas probabilidades são obtidas a partir da utilização da rede neural *BlazeFace*. A partir de testes, definiu-se que o estado de sonolência é detectado em casos onde a média dessas probalidades é inferior a 60%.

A abordagem utilizada para disparar o aviso que o motorista está em estado de sonolência é similar à utilizada por YU *et al.* (2019). Neste caso, é mantido um contador que é atualizado a cada *frame*. Esse contador é incrementado, caso seja detectado sinal de sonolência no *frame* atual. O contador é decrementado caso não seja detectado um sinal de sonolência. Quando o contador ultrapassar o valor de 5 é emitido um sinal de aviso, ou seja, o sinal é disparado após o motorista apresentar sinais de sonolência em 5 *frames* em sequência.

5 IMPLEMENTAÇÃO DESENVOLVIDA E RESULTADOS OBTIDOS

A solução foi desenvolvida de forma a ser executada no próprio *smartphone* do motorista, evitando-se assim a necessidade de sensores e outros dispositivos de *hardware* com um custo elevado. Como pode ser observado Figura 21, o sistema operacional *Android* lidera o mercado de *smartphones*, porém o sistema operacional *iOS* também apresenta uma elevada fatia do mercado. Desta forma, torna-se interessante que a solução seja desenvolvida para ser utilizada em ambos os sistemas operacionais.

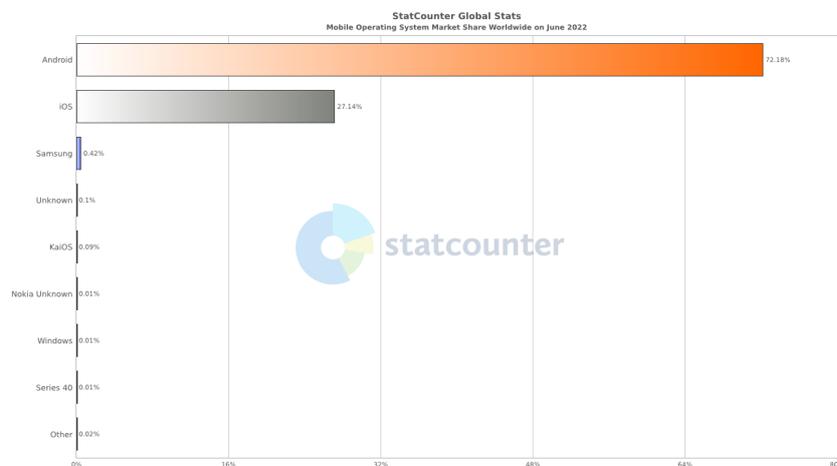


Figura 21 – Fatia de mercado de cada um dos sistemas operacionais para dispositivos móveis (Jun 2022)

Fonte: (STATCOUNTER, 2022)

Existem basicamente três abordagens para o desenvolvimento de aplicativos para as plataformas *Android* e *iOS*. A primeira delas consiste em desenvolver as duas aplicações em código nativo. Por exemplo, no desenvolvimento para a plataforma *Android*, seriam utilizadas as linguagens de programação *Kotlin* ou *Java* e as funções próprias do sistema operacional *Android*. Já para o desenvolvimento para a plataforma *iOS*, seriam utilizadas as linguagens *Swift* ou *Objective-C* e as funções nativas do sistema operacional *iOS*. A principal desvantagem dessa abordagem está relacionada ao custo e ao tempo de desenvolvimento, visto que seria necessário desenvolver duas aplicações com códigos diferentes. Além disso, os custos de manutenção também aumentam, uma vez que as atualizações devem ser realizadas em ambos os códigos (JUNTUNEN; JALONEN; LUUKKAINEN, 2013).

Outra possível abordagem seria a utilização de *WebApps*, que consiste em desenvolver a aplicação como uma página *web* e executá-la em um navegador. Essa abordagem é inviável neste trabalho, visto que não seria possível ter acesso aos recursos nativos do *smartphone*, como por exemplo, a câmera (TAPPAREL, 2013).

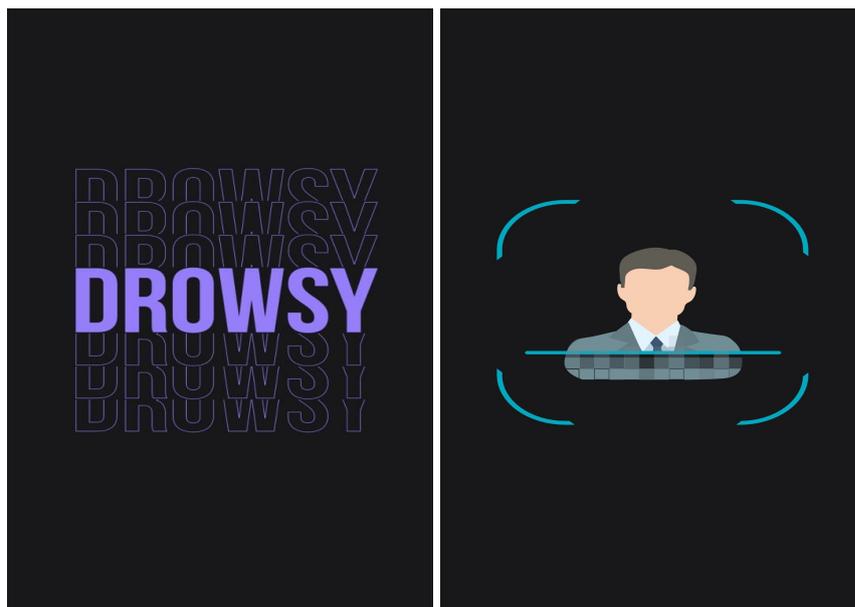
Por fim, uma terceira abordagem consiste em utilizar um *framework* para o desenvolvimento de aplicações multiplataforma. Neste caso, é desenvolvido um único código-fonte, que é compilado em código nativo para os diferentes sistemas operacionais. Essa abordagem tem como principal vantagem uma redução nos tempos de desenvolvimento e de manutenção, visto que todo o desenvolvimento é realizado a partir de uma mesma base de código (MAHARANA, 2017).

Para o desenvolvimento do aplicativo móvel proposto neste trabalho, optou-se pela utilização de um *framework* para o desenvolvimento de aplicações multiplataforma. Mais especificamente foi utilizado o *framework React Native*, que é um *framework* de código aberto, desenvolvido pela empresa *Meta* (NATIVE, 2022). Neste, uma única aplicação é desenvolvida utilizando a linguagem de programação *JavaScript* e, em tempo de execução, o *React Native* converte o código correspondente em código nativo, seja para a plataforma *Android* ou *iOS*.

Para a implementação das operações de processamento de imagens do aplicativo foram utilizadas duas bibliotecas. Para execução do algoritmo de *Haar* (*Viola Jones*), foi utilizada a biblioteca *Pico* (*Pixel Intensity Comparison-based Object detection*), que é uma biblioteca código aberto que foi desenvolvida por MARKUŠ *et al.* (2013). Essa foi desenvolvida em 2013 para a linguagem de programação C e, posteriormente, em 2018, foi implementada utilizando a linguagem *JavaScript*. Para a utilização da rede neural do tipo *BlazeFace*, foi utilizada a biblioteca *ML Kit*, desenvolvida pelo *Google*. Essa biblioteca implementa diversos algoritmos de aprendizado de máquina para serem executados em dispositivos móveis. Essa pode ser utilizada de forma gratuita para o desenvolvimento de aplicativos tanto para as plataformas *Android* quanto *iOS*.

5.1 INTERFACE DO APLICATIVO DESENVOLVIDO

No desenvolvimento de um aplicativo, a fim de oferecer uma usabilidade intuitiva, é importante definir padrões que devem ser utilizados em todo o desenvolvimento, são eles: cores, ícones, logo, fontes e espaçamentos. Para reduzir a fadiga ocular em casos de uso prolongado, principalmente a noite, optou-se pela utilização de uma paleta de cores e um tema escuro. Os ícones foram utilizados seguindo a biblioteca *Material Design Icons*, desenvolvida pela *Google*. A fonte utilizada foi a *Roboto* que é disponibilizada pelo *Google Fonts*. O logo foi desenvolvido com base no nome do aplicativo (*Drowsy*). Os espaçamentos entre os elementos são amplos de forma a facilitar a distinção entre os elementos da tela. Ao abrir o aplicativo tem-se a tela inicial com o logo da aplicação (Figura 22 (a)). Após, é realizado o carregamento dos modelos dos classificadores *Haar* e *BlazeFace*. Durante o período de carregamento dos modelos é exibida a tela apresentada na Figura 22 (b).



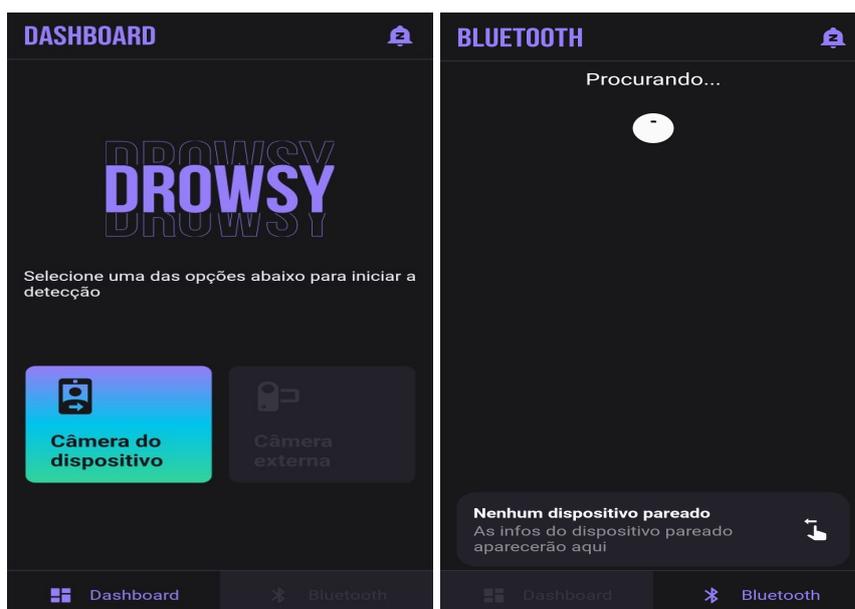
a) Tela principal

b) Tela de carregamento

Figura 22 – Telas iniciais do aplicativo

Fonte: O autor

Após o carregamento, é exibida a tela de entrada, que é apresentada na Figura 23 (a). Nesta tela é possível selecionar a funcionalidade de detecção facial, utilizando o botão "Câmera do dispositivo". Além disso, foi criado um botão de "Câmera externa" e a tela de *Bluetooth*, como pode ser observado na Figura 23 (b). Essas opções foram inseridas para implementações futuras, caso seja necessário parear com uma câmera externa.



a) Tela de *dashboard*

b) Tela de *bluetooth*

Figura 23 – Telas principais do aplicativo

Fonte: O autor

No momento que a opção de detecção facial é selecionada, tem-se a abertura da câmera, sendo sinalizada a área correspondente a face do motorista. Nesta tela, é possível verificar se os olhos encontram-se fechados, como pode ser observado na Figura 24 (a). Quando a pontuação, apresentada na tela, ultrapassar o valor de 4, a área da face é destacada em vermelho e o dispositivo começa a vibrar. O dispositivo só para de vibrar quando for detectado que os olhos encontram-se abertos.

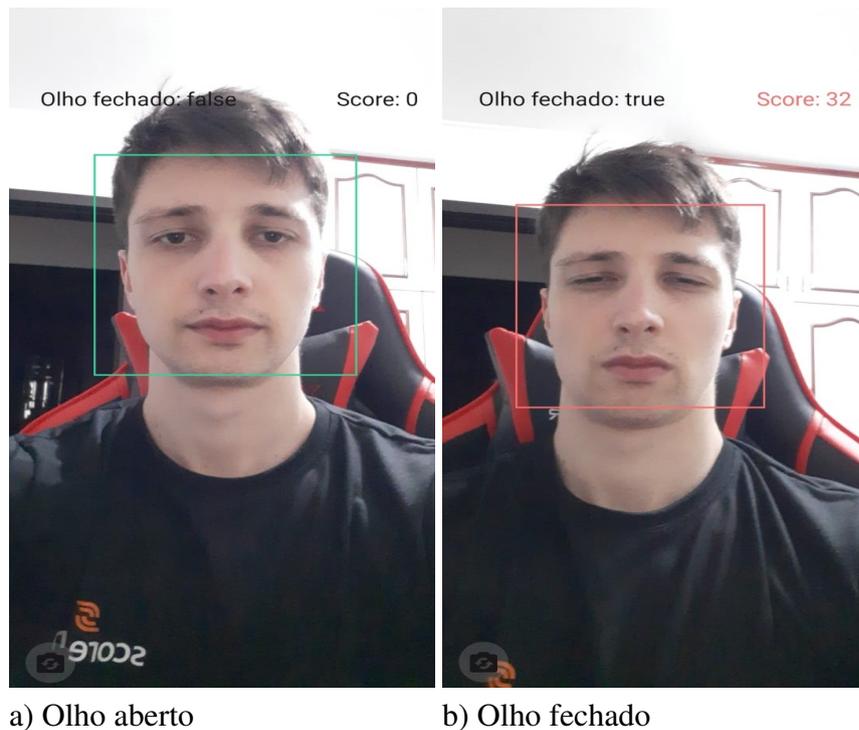


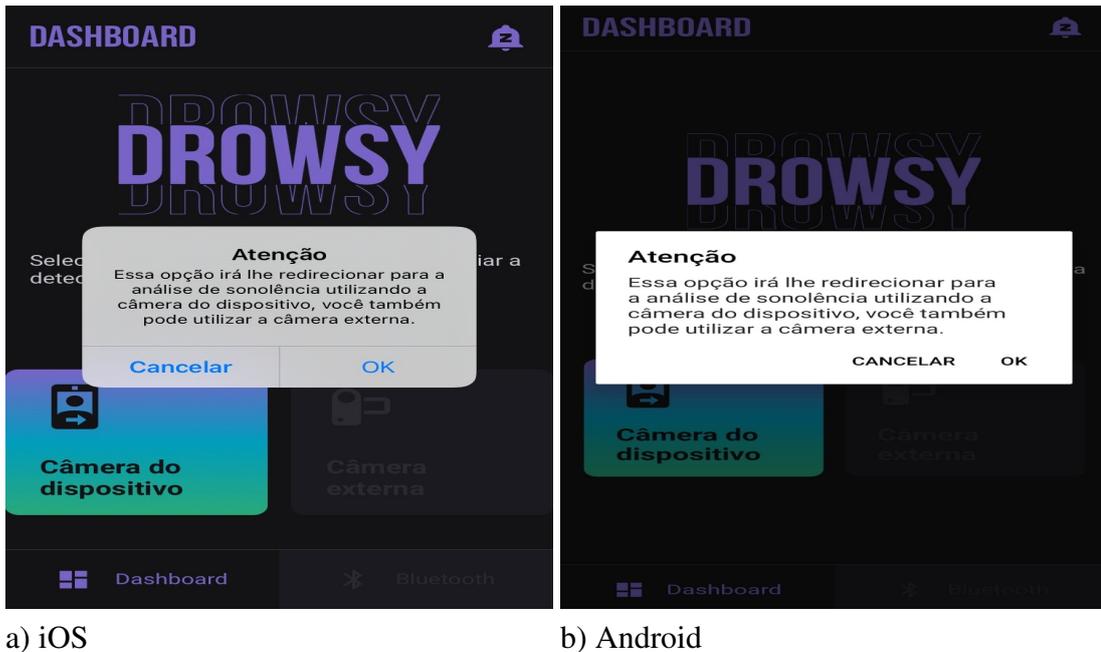
Figura 24 – Telas de detecção da face e classificação

Fonte: O autor

5.2 TESTES NAS PLATAFORMAS ANDROID E IOS

Primeiramente, foram realizados testes procurando verificar o correto funcionamento da implementação desenvolvida nas plataformas *Android* e *iOS*. Os testes na plataforma *Android* foram realizados utilizando um *smartphone Galaxy M20* da *Samsung*. Esse *smartphone* possui um processador *Exynos 7904* com 8 núcleos de até 1,8 GHz e uma memória *RAM* de 4 GB. A câmera traseira possui uma resolução de 13 *megapixels* e abertura de F1.9 e a câmera frontal possui uma resolução de 5 *megapixels*. O sistema operacional utilizado é o *Android 10*. Os testes no sistema operacional *iOS* foram realizados utilizando um *smartphone iPhone 8* da *Apple*. Esse *smartphone* possui 2 GB de memória *RAM* e um processador *Apple A11 Bionic* com 6 núcleos de até 2.39 GHz. A câmera traseira possui uma resolução de 12 *megapixels* e abertura de F1.8 e a câmera frontal possui uma resolução de 7 *megapixels*. O sistema operacional utilizado foi o *iOS 15*.

O aplicativo desenvolvido funcionou de forma correta em ambas as plataformas. Pequenas diferenças foram observadas em animações de troca de telas, abas, botões e mensagens de alerta. Essas diferenças devem-se a diferenças existentes entre os componentes nativos utilizados. Por exemplo, a tela de *dashboard*, Figura 23 (a), apresenta pequenas diferenças devido ao comportamento nativo dos componentes, como por exemplo, a animação ao clicar nos botões e a mensagem de alerta, como pode ser observado na Figura 25.



a) iOS

b) Android

Figura 25 – Mensagem de alerta

Fonte: O autor

5.3 TESTES DE DESEMPENHO

Um dos principais requisitos da implementação desenvolvida é o funcionamento em tempo real com o mínimo de atraso. Para uma primeira análise do desempenho da implementação foi considerado o tempo necessário para a execução da etapa de detecção e classificação de abertura dos olhos um único *frame*. Neste primeiro teste a classificação dos olhos foi realizada utilizando a imagem completa, ou seja, sem a realização da etapa de reconhecimento da face.

A partir destes testes verificou-se que o tempo total de processamento era superior à 1 segundo (1052 ms), no *smartphone* com a plataforma *Android*. Esse baixo desempenho era decorrente das dimensões da imagem de entrada, que apresentava uma resolução de 2448×3264 *pixels*. A partir de um redimensionamento da imagem de entrada para uma resolução de 256×341 *pixels* foi possível reduzir o tempo de execução para cerca de 500 ms, sem provocar um aumento nos erros de detecção.

Posteriormente, foi adicionada a etapa de detecção facial com o algoritmo de *Viola Jones*. A detecção da face apresentou um tempo de execução de aproximadamente 90 ms e provocou melhorias significativas nas etapas posteriores, visto que reduziu a área de interesse para a detecção e classificação dos olhos. De fato, a etapa de detecção e classificação dos olhos passou a ser executada em aproximadamente 150 ms. Assim, o tempo total para a execução de todas as etapas de processamento de um *frame* foi de aproximadamente 350 ms, sendo possível a avaliação de quase 3 *frames* por segundo. Na Tabela 2 tem-se um resumo dos tempos de execução de todas as etapas de processamento. Os valores foram obtidos considerando a média dos tempos obtidos em 50 *frames*.

Tabela 2 – Etapas e seus desempenhos em cada uma das plataformas

Etapa	Tempo <i>Android</i>	Tempo <i>iOS</i>
Captura e redimensionamento das imagens	107 ms	328 ms
Conversão para tons de cinza e detecção da face	86 ms	60 ms
Detecção e classificação dos olhos	150,5 ms	41 ms
Tempo Total	343,5	429

5.4 TESTES EM AMBIENTES COM ALTA LUMINOSIDADE

Foram realizados testes considerando ambientes com uma alta luminosidade. Para tanto, o *smartphone* foi posicionado em um suporte, e juntamente dele, foi posicionada uma lâmpada de luz branca com seu feixe de luz apontado diretamente para a face do motorista, como pode ser observado na Figura 26.



a) Lâmpada desligada

b) Lâmpada ligada

Figura 26 – Estrutura para simular ambiente de alta luminosidade

Fonte: O autor

Observou-se que a implementação desenvolvida apresentou um desempenho satisfatório em ambientes de alta luminosidade, visto que todas as etapas de processamento foram executadas sem erros de detecção (Figura 27). O único problema detectado foi devido ao ajuste do foco da câmera. Em alguns casos, a câmera alternava o foco diretamente para a lâmpada, escurecendo o restante da imagem. Esse problema foi resolvido ajustando o foco da câmera no início do processamento.



Figura 27 – Imagem capturada com alta luminosidade

Fonte: O autor

5.5 TESTES EM AMBIENTES COM BAIXA LUMINOSIDADE

Foram realizados ainda testes procurando simular um ambiente de condução noturna, onde as condições de luminosidade são baixas. Neste caso, o *smartphone* foi posicionado no suporte e todas as luzes foram desligadas. A partir dos testes, verificou-se que a etapa de detecção da face foi realizada sem erros, como pode ser observado na Figura 28. No entanto, a etapa de classificação dos olhos foi impactada significativamente, de modo que em todos *frames* foi identificado que os olhos estavam fechados. Para a solução deste problema, seria necessário a utilização de um *smartphone* ou dispositivo que tenha uma câmera com sensor de modo de visão noturna (sensor *IR (infravermelho)*).

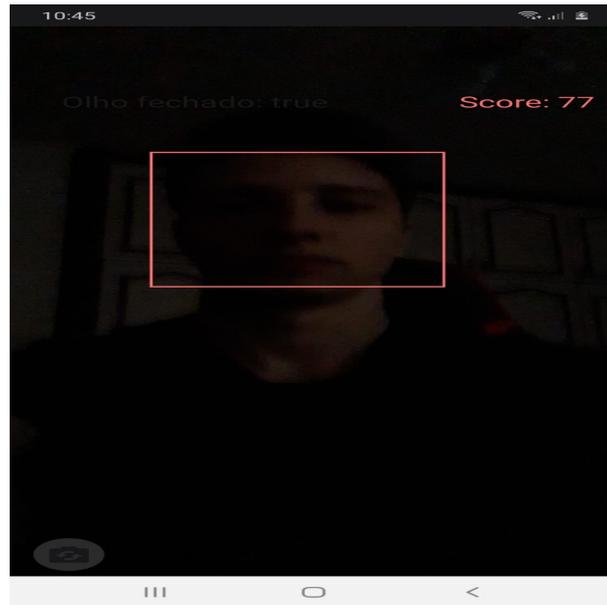
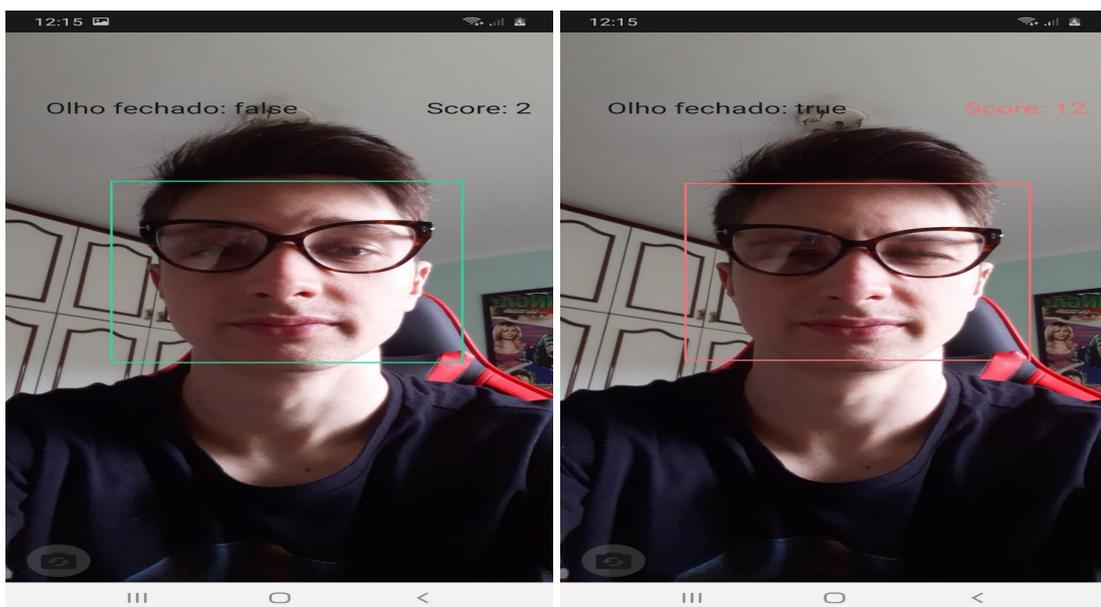


Figura 28 – Imagem capturada com baixa luminosidade

Fonte: O autor

5.6 MOTORISTA QUE UTILIZA ÓCULOS

Foram realizados ainda testes procurando verificar o desempenho do sistema em casos onde o motorista utiliza óculos (Figura 29). A partir desses, testes verificou-se que a utilização de óculos não impacta em nenhuma das etapas de processamento. Desta forma, a classificação de sonolência é realizada de forma precisa.



a) Olhos abertos

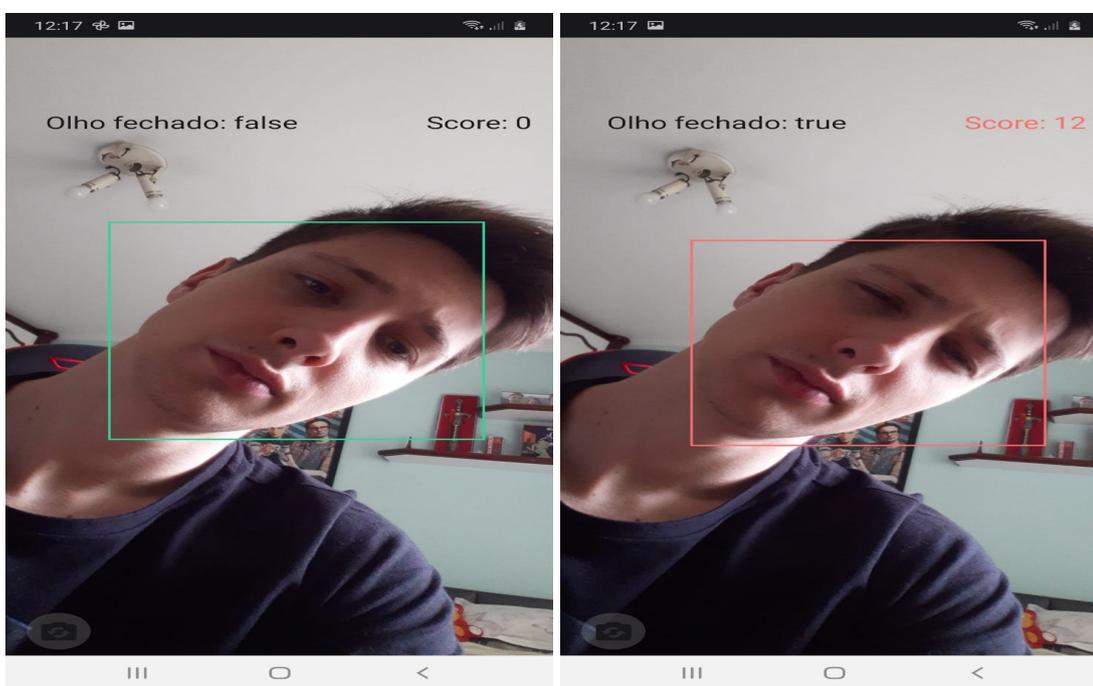
b) Olhos fechados

Figura 29 – Motorista com óculos

Fonte: O autor

5.7 MAL POSICIONAMENTO DA CÂMERA

Para a simulação de um mal posicionamento da câmera, foram realizados testes posicionando a câmera com diferentes ângulos de inclinação. Observou-se que a implementação desenvolvida funcionou corretamente para inclinação com ângulos até 45°. Em casos, onde o ângulo de inclinação seja maior que 45°, a face não foi detectada de forma correta, causando erros no sistema de detecção de sonolência. Foi implementado ainda uma funcionalidade que procura corrigir erros em casos onde *smartphone* tenha sido posicionado de cabeça para baixo. Neste caso, quando ocorre uma falha de detecção, o próximo *frame* é rotacionado em 180°. Caso a face seja detectada, essa rotação é mantida nos *frames* subsequentes. Caso a face não seja detectada, a rotação não é realizada nos próximos *frames*.



a) Olhos abertos

b) Olhos fechados

Figura 30 – Imagem capturada com uma inclinação de 45°

Fonte: O autor

Para situações em que o rosto está de cabeça para baixo na imagem de entrada, foi adicionada a tratativa de executar uma rotação de 180° nos próximos *frames*, corrigindo assim um possível erro de detecção, ver Figura 31. De qualquer forma, o recomendado para extrair melhor desempenho é iniciar a detecção com o rosto alinhado, de forma que na imagem de entrada o topo da cabeça esteja alinhado ao topo da imagem.

5.8 PARTE DA FACE ENCOBERTA

Por fim, foram realizados testes para verificar o impacto de algum objeto encobrindo parte da face do motorista. Nestes testes, foi utilizada a mão para cobrir, primeiramente a boca,

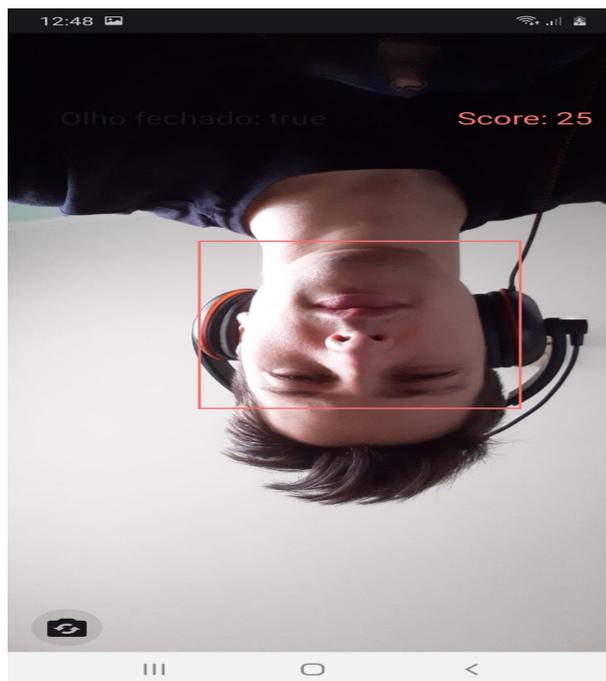
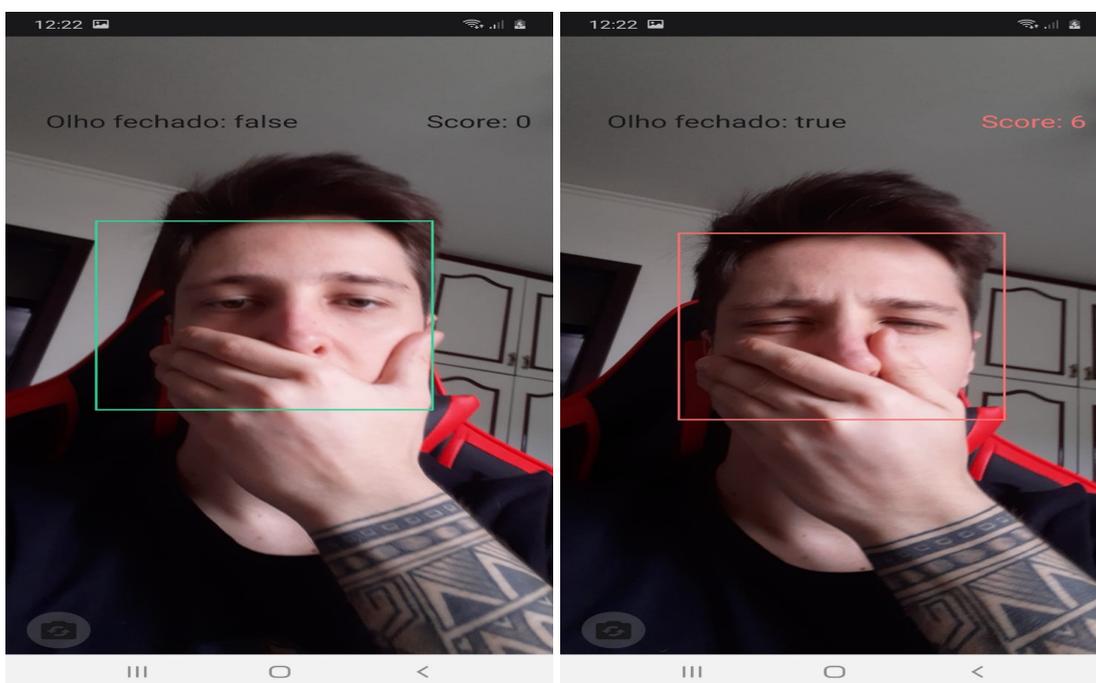


Figura 31 – Exemplo de imagem capturada em que o rosto está de cabeça para baixo

Fonte: O autor

depois um dos olhos, parte do nariz e da boca. Os testes realizados mostraram que o sistema funciona corretamente em casos onde uma parte da face encontra-se encoberta (Figura 32).

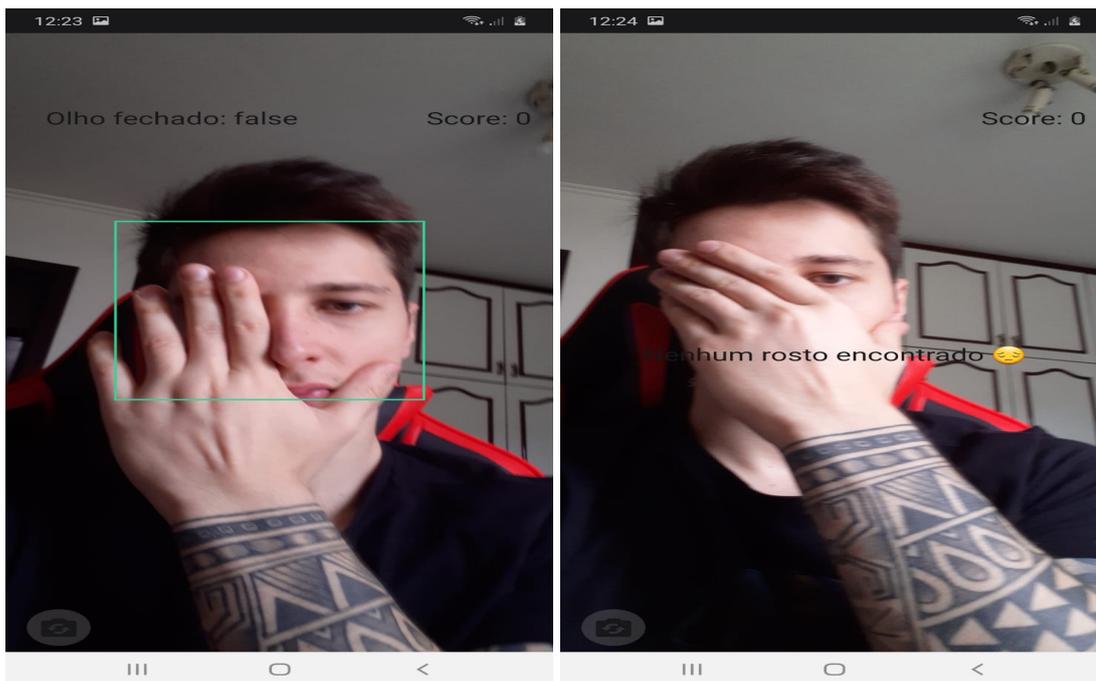


a) Olhos abertos

b) Olhos fechados

Figura 32 – Exemplo de imagem capturada em testes com a mão encobrendo a boca

Fonte: O autor



a) Olho

b) Metade da face

Figura 33 – Exemplo de imagem capturada em testes com a mão encobrindo regiões da face

Fonte: O autor

A mão sobre a face não impacta na detecção do sistema, como pode ser visto no caso da mão sobre a boca do motorista, ver Figura 32, e sobre um dos olhos, ver Figura 33 (a). Portanto quando o motorista tiver a mão sobre parte do rosto, não causará impacto na detecção do sistema.

Observou-se problemas somente em casos, onde mais de uma das característica do rosto encontra-se encoberta, como por exemplo, quando a boca, o nariz e um dos olhos está encoberto, ver Figura 33 (b). No entanto, essa situação não é preocupante, uma vez esse não é um comportamento comum durante a direção de um veículo.

6 CONSIDERAÇÕES FINAIS

Neste trabalho foi desenvolvido um sistema não intrusivo e de baixo custo para a detecção de sonolência em motoristas. Esse se baseia na análise de imagens obtidas da face do motorista e foi desenvolvido de forma a ser executado no próprio *smartphone* do motorista, sem a necessidade de uso de sensores ou dispositivos de *hardware* com um custo mais elevado.

A solução foi desenvolvida para os sistemas operacionais *Android* e *iOS*. Para tanto, utilizou-se o *framework React Native*, que é um *framework* para o desenvolvimento de aplicativos multiplataforma. A utilização deste *framework*, mostrou-se adequada, pois possibilitou a utilização de uma única base de código para ambas as plataformas e reduziu significativamente o tempo de desenvolvimento. Os testes realizados mostram que a aplicação desenvolvida funcionou de forma adequada em ambas as plataformas.

A solução desenvolvida tem início a captura das imagens, que é realizada utilizando uma câmera de um *smartphone* posicionada sobre o painel do veículo. As imagens foram capturadas em um formato colorido e, posteriormente, redimensionadas a fim de otimizar as etapas posteriores. No redimensionamento da imagem, optou-se por uma largura fixa de 256 *pixels* e a altura foi redimensionada de forma a mantêr as proporções da imagem de entrada sem distorções. A partir de testes, verificou-se que esse redimensionamento causou um ganho significativo de desempenho, sem prejudicar a precisão do sistema. Antes do redimensionamento, a etapa de classificação dos olhos foi executada diretamente sobre a imagem de entrada (uma imagem de 2448×3264 *pixels*), apresentando um tempo de execução de aproximadamente 1 segundo (1052 ms), em um *smartphone Galaxy M20*. Após o redimensionamento, essa etapa passou a ser executada em menos de 500 ms.

Posteriormente, é realizado um procedimento de conversão da imagem para tons de cinza e executado o método de detecção facial. Para a detecção da face foi utilizado um Classificador *Haar*. Esse classificador apresentou resultados satisfatórios, com um baixo custo de processamento. O classificador apresentou um tempo médio de execução de 86 ms em testes utilizando um *smartphone Galaxy M20* e 60 ms em testes realizados em um *iPhone 8*.

Após a detecção da face, a região em torno da face é utilizada como entrada para a etapa de classificação dos olhos em abertos ou fechados, conforme *PERCLOS*. Para a classificação dos olhos optou-se pela utilização de uma rede neural do tipo *BlazeFace*. Essa rede neural foi desenvolvida especificamente para ser executado em dispositivos móveis. A rede neural *BlazeFace* apresentou resultados satisfatórios, com um baixo tempo de execução. Essa apresentou uma precisão de 98,61% em ambientes com iluminação normal e sem obstáculos ocultando mais de uma característica da face. O tempo de execução médio foi de 150,51 ms em um *Galaxy M20* e de 41 ms em um *iPhone 8*.

Para a validação da solução foram realizados testes em diversos ambientes e condições de luminosidade. Observou-se que o sistema apresenta erros de classificação em situações onde o *smartphone* encontra-se posicionado de forma incorreta (inclinação superior à 45°). Para casos onde o *smartphone* encontra-se de cabeça para baixo, foi implementada uma funcionalidade que rotaciona o próximo *frame* em 180°. Caso a face seja detectada, essa rotação é mantida para os próximos *frames*.

Observou-se ainda que o sistema pode apresentar erros de detecção em ambientes com baixa luminosidade, como por exemplo, em imagens capturadas a noite, onde as imagens apresentam baixa nitidez. Esse problema pode ser facilmente resolvido através da utilização de uma câmera infravermelho, não sendo necessária nenhuma alteração na solução. A utilização de uma câmera infravermelho externa pode ser realizada através da utilização da funcionalidade de *bluetooth* implementada, bastando adicionar a câmera na lista de dispositivos suportados.

No que se refere ao tempo de execução, o sistema apresentou um tempo total de processamento de um *frame* de 343,5 ms em um *Galaxy M20* e 429 ms em *iPhone 8*. Desta forma, conclui-se que a solução proposta é viável para ser executada em *smartphones* com configurações intermediárias, visto que é possível analisar de 2 a 3 *frames* por segundo.

6.1 TRABALHOS FUTUROS

Como propostas para trabalhos futuros, sugere-se

- Uso da função de *bluetooth*, que já foi implementada, para a utilização de uma câmera externa, possibilitando utilizar o *smartphone* somente para o processamento das imagens. Isso possibilitaria a utilização de uma câmera infravermelho externa e, conseqüentemente, solucionaria o problema de detecções em ambientes de baixa iluminação.
- Implementação de outras métricas para análise da sonolência, além da *PERCLOS*. Por exemplo, poderia ser utilizada a variação da inclinação da cabeça entre *frames*. Essa variação poderia indicar situações onde o motorista está entrando em estado de sonolência e seu pescoço está perdendo a rigidez.

REFERÊNCIAS

- ABRAMET. **Problemas na saúde de motoristas causaram mais de 280 mil acidentes nas rodovias desde 2014, aponta Abramet**. 2020. Disponível em: <<https://www.abramet.com.br/noticias>>. [Online; acessado 09 de março de 2022].
- AISIN. **Aisin develops new products for enabling Advanced Driving Assist Technology**. 2021. Disponível em: <<https://www.aisin.com/en/news/2021/005406.html>>. [Online; acessado 03 de março de 2022].
- AKIN, M. *et al.* Estimating vigilance level by using eeg and emg signals. **Neural Computing and Applications**, Springer, v. 17, n. 3, p. 227–236, 2008.
- ALIOUA, N.; AMINE, A.; RZIZA, M. Driver's fatigue detection based on yawning extraction. **International journal of vehicular technology**, Hindawi, v. 2014, 2014.
- ALSHAQAQI, B. *et al.* Driver drowsiness detection system. In: IEEE. **2013 8th International Workshop on Systems, Signal Processing and their Applications (WoSSPA)**. [S.l.], 2013. p. 151–155.
- ARAKAWA, T. Trends and future prospects of the drowsiness detection and estimation technology. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 21, n. 23, p. 7921, 2021.
- BAEK, H. J. *et al.* A smart health monitoring chair for nonintrusive measurement of biological signals. **IEEE transactions on Information Technology in Biomedicine**, IEEE, v. 16, n. 1, p. 150–158, 2011.
- BAZAREVSKY, V. *et al.* Blazeface: Sub-millisecond neural face detection on mobile gpus. **arXiv preprint arXiv:1907.05047**, 2019.
- BERGASA, L. M. *et al.* Real-time system for monitoring driver vigilance. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 7, n. 1, p. 63–77, 2006.
- BRADSKI, G.; KAEHLER, A. **Learning OpenCV: Computer vision with the OpenCV library**. [S.l.]: "O'Reilly Media, Inc.", 2008.
- BYEON, M.-K. *et al.* A study of hrv analysis to detect drowsiness states of drivers. In: **Proceedings of the 24th IASTED international conference on Biomedical engineering**. [S.l.: s.n.], 2006. p. 153–155.
- CHACON-MURGUIA, M. I.; PRIETO-RESENDIZ, C. Detecting driver drowsiness: A survey of system designs and technology. **IEEE Consumer Electronics Magazine**, IEEE, v. 4, n. 4, p. 107–119, 2015.
- DASGUPTA, A.; RAHMAN, D.; ROUTRAY, A. A smartphone-based drowsiness detection and warning system for automotive drivers. **IEEE transactions on intelligent transportation systems**, IEEE, v. 20, n. 11, p. 4045–4054, 2018.
- DERAM, P. Vehicle based detection of inattentive driving for integration in an adaptive lane departure warning system. Citeseer, 2004.

DINGES, D. F. *et al.* **Evaluation of techniques for ocular measurement as an index of fatigue and as the basis for alertness management.** [S.l.], 1998.

DINGES D.F.; MALLIS, M. M. G. P. J. N. **Evaluation of Techniques for Ocular Measurement as an Index of Fatigue and the Basis for Alertness Management.** 1998. Disponível em: <<https://ntlrepository.blob.core.windows.net/lib/21000/21900/21955/PB99150237.pdf>>. [Online; acessado 03 de março de 2022].

EDWARDS, D. J. *et al.* Evaluation of fatigue management technologies using weighted feature matrix method. In: **Proceedings of the Fourth International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design.** [S.l.: s.n.], 2007. p. 146–152.

ERIKSSON, M.; PAPANIKOTOPOULOS, N. P. Eye-tracking for detection of driver fatigue. In: IEEE. **Proceedings of Conference on Intelligent Transportation Systems.** [S.l.], 1997. p. 314–319.

FENG, R.; ZHANG, G.; CHENG, B. An on-board system for detecting driver drowsiness based on multi-sensor data fusion using dempster-shafer theory. In: IEEE. **2009 international conference on networking, sensing and control.** [S.l.], 2009. p. 897–902.

FORSMAN, P. M. *et al.* Efficient driver drowsiness detection at moderate levels of drowsiness. **Accident Analysis & Prevention**, Elsevier, v. 50, p. 341–350, 2013.

GÓMEZ-CLAPERS, J.; CASANELLA, R. A fast and easy-to-use ecg acquisition and heart rate monitoring system using a wireless steering wheel. **IEEE Sensors Journal**, IEEE, v. 12, n. 3, p. 610–616, 2011.

GRACE, R.; STEWARD, S. Drowsy driver monitor and warning system. In: **International driving symposium on human factors in driver assessment, training and vehicle design.** [S.l.: s.n.], 2001. v. 8, p. 201–208.

HANOWSKI, R. N. **The Drowsy Driver Warning System Field Operational Test: Data Collection Methods.** 2008. Disponível em: <<https://vtechworks.lib.vt.edu/bitstream/handle/10919/55072/810035.pdf>>. [Online; acessado 03 de março de 2022].

HAYASHI, K. *et al.* Individualized drowsiness detection during driving by pulse wave analysis with neural network. In: IEEE. **Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005.** [S.l.], 2005. p. 901–906.

HOWARD, A. G. *et al.* Mobilenets: Efficient convolutional neural networks for mobile vision applications. **arXiv preprint arXiv:1704.04861**, 2017.

HU, S.; ZHENG, G. Driver drowsiness detection with eyelid related parameters by support vector machine. **Expert Systems with Applications**, Elsevier, v. 36, n. 4, p. 7651–7658, 2009.

INTHANON, P.; MUNGSING, S. Detection of drowsiness from facial images in real-time video media using nvidia jetson nano. In: **2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON).** [S.l.: s.n.], 2020. p. 246–249.

Jl, Q.; ZHU, Z.; LAN, P. Real-time nonintrusive monitoring and prediction of driver fatigue. **IEEE transactions on vehicular technology**, IEEE, v. 53, n. 4, p. 1052–1068, 2004.

- JOHN, M. S.; RISSER, M. R.; KOBUS, D. A. Toward a usable closed-loop attention management system: Predicting vigilance from minimal contact head, eye, and eeg measures. **Proceedings of the 2nd Annual Augmented Cognition, San Francisco, CA**, Citeseer, p. 12–18, 2006.
- JUNTUNEN, A.; JALONEN, E.; LUUKKAINEN, S. Html 5 in mobile devices—drivers and restraints. In: IEEE. **2013 46th Hawaii International Conference on System Sciences**. [S.l.], 2013. p. 1053–1062.
- KATIYAR, S. K.; ARUN, P. Comparative analysis of common edge detection techniques in context of object extraction. **arXiv preprint arXiv:1405.6132**, 2014.
- KAZEMI, V.; SULLIVAN, J. One millisecond face alignment with an ensemble of regression trees. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2014. p. 1867–1874.
- KHUSHABA, R. N. *et al.* Driver drowsiness classification using fuzzy wavelet-packet-based feature-extraction algorithm. **IEEE transactions on biomedical engineering**, IEEE, v. 58, n. 1, p. 121–131, 2010.
- KIM, Y.; KIM, Y.; HAHN, M. Detecting driver fatigue based on the driver’s response pattern and the front view environment of an automobile. In: IEEE. **2008 Second International Symposium on Universal Communication**. [S.l.], 2008. p. 237–240.
- KLINGEBERG, T.; SCHILLING, M. Mobile wearable device for long term monitoring of vital signs. **Computer methods and programs in biomedicine**, Elsevier, v. 106, n. 2, p. 89–96, 2012.
- KOBAYASHI, H. Emg/ecg acquisition system with online adjustable parameters using zigbee wireless technology. **Electronics and Communications in Japan**, Wiley Online Library, v. 96, n. 5, p. 1–10, 2013.
- KURT, M. B. *et al.* The ann-based computing of drowsy level. **Expert Systems with Applications**, Elsevier, v. 36, n. 2, p. 2534–2542, 2009.
- KWAPISZ, J. R.; WEISS, G. M.; MOORE, S. A. Activity recognition using cell phone accelerometers. **ACM SigKDD Explorations Newsletter**, ACM New York, NY, USA, v. 12, n. 2, p. 74–82, 2011.
- LIN, C.-T. *et al.* Adaptive eeg-based alertness estimation system by using ica-based fuzzy neural networks. **IEEE Transactions on Circuits and Systems I: Regular Papers**, IEEE, v. 53, n. 11, p. 2469–2476, 2006.
- LIN, F.-C. *et al.* Generalized eeg-based drowsiness prediction system by using a self-organizing neural fuzzy system. **IEEE Transactions on Circuits and Systems I: Regular Papers**, IEEE, v. 59, n. 9, p. 2044–2055, 2012.
- LIU, C. C.; HOSKING, S. G.; LENNÉ, M. G. Predicting driver drowsiness using vehicle measures: Recent insights and future challenges. **Journal of safety research**, Elsevier, v. 40, n. 4, p. 239–245, 2009.

- MAHARANA, N. **Cross Platform Mobile Apps and Its Pros and Cons**. 2017. Disponível em: <<https://medium.com/@netranandamaharana/cross-platform-mobile-apps-and-its-pros-and-cons-9c257ec64e94>>. [Online; acessado 20 de junho de 2022].
- MARKUŠ, N. *et al.* Object detection with pixel intensity comparisons organized in decision trees. **arXiv preprint arXiv:1305.4537**, 2013.
- MARSHALL, S. P.; PLEYDELL-PEARCE, C. W.; DICKSON, B. T. Integrating psychophysiological measures of cognitive workload and eye movements to detect strategy shifts. In: IEEE. **36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the**. [S.l.], 2003. p. 6–pp.
- MICHAIL, E. *et al.* Eeg and hrv markers of sleepiness and loss of control during car driving. In: IEEE. **2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society**. [S.l.], 2008. p. 2566–2569.
- MURATA, A.; IWASE, H. Evaluation of mental workload by fluctuation analysis of pupil area. In: IEEE. **Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Vol. 20 Biomedical Engineering Towards the Year 2000 and Beyond (Cat. No. 98CH36286)**. [S.l.], 1998. v. 6, p. 3094–3097.
- NATIVE, R. **React Native**. 2022. Disponível em: <<https://reactnative.dev/>>. [Online; acessado 20 de junho de 2022].
- OMS. **Road traffic injuries**. 2021. Disponível em: <<https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries>>. [Online; acessado 12 de março de 2022].
- OTMANI, S. *et al.* Effect of driving duration and partial sleep deprivation on subsequent alertness and performance of car drivers. **Physiology & behavior**, Elsevier, v. 84, n. 5, p. 715–724, 2005.
- PARIKH, P.; MICHELI-TZANAKOU, E. Detecting drowsiness while driving using wavelet transform. In: IEEE. **IEEE 30th Annual Northeast Bioengineering Conference, 2004. Proceedings of the**. [S.l.], 2004. p. 79–80.
- PATEL, M. *et al.* Applying neural network analysis on heart rate variability data to assess driver fatigue. **Expert systems with Applications**, Elsevier, v. 38, n. 6, p. 7235–7242, 2011.
- SAHAYADHAS, A.; SUNDARAJ, K.; MURUGAPPAN, M. Detecting driver drowsiness based on sensors: a review. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 12, n. 12, p. 16937–16953, 2012.
- SAINI, V.; SAINI, R. Driver drowsiness detection system and techniques: a review. **International Journal of Computer Science and Information Technologies**, Citeseer, v. 5, n. 3, p. 4245–4249, 2014.
- SAMSUNG. **Samsung Galaxy M20**. 2019. Disponível em: <<https://www.samsung.com/levant/smartphones/galaxy-m20/buy/iraq/#camera>>. [Online; acessado 04 de abril de 2022].
- SATO, N. *et al.* Power spectral analysis of heart rate variability in type a females during a psychomotor task. **Journal of psychosomatic research**, Elsevier, v. 45, n. 2, p. 159–169, 1998.

STATCOUNTER. **Mobile Operating System Market Share Worldwide**. 2022. Disponível em: <<https://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-202206-202206-bar>>. [Online; acessado 20 de junho de 2022].

TABRIZI, P. R.; ZOROOFI, R. A. Drowsiness detection based on brightness and numeral features of eye image. In: IEEE. **2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing**. [S.l.], 2009. p. 1310–1313.

TANAKA, Y. Arousal level and blink activity. **Shinrigaku kenkyu: The Japanese Journal of Psychology**, v. 70, n. 1, p. 1–8, 1999.

TAPPAREL, Y. **Touch-Optimised Mobile Interface for Invenio Digital Library**. Tese (Doutorado) — Ecole Polytechnique, Lausanne, 2013.

TIPPRASERT, W. *et al.* A method of driver's eyes closure and yawning detection for drowsiness analysis by infrared camera. In: **2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP)**. [S.l.: s.n.], 2019. p. 61–64.

UENO, H.; KANEDA, M.; TSUKINO, M. Development of drowsiness detection system. In: IEEE. **Proceedings of VNIS'94-1994 Vehicle Navigation and Information Systems Conference**. [S.l.], 1994. p. 15–20.

VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. In: IEEE. **Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001**. [S.l.], 2001. v. 1, p. I–I.

VURAL, E. Video based detection of driver fatigue. Mühendislik ve Fen Bilimleri Enstitüsü, 2009.

VURAL, E. *et al.* Discrimination of moderate and acute drowsiness based on spontaneous facial expressions. In: **2010 20th International Conference on Pattern Recognition**. [S.l.: s.n.], 2010. p. 3874–3877.

WELLS, P. Deaths and injuries from car accidents: an intractable problem? **Journal of Cleaner Production**, Elsevier, v. 15, n. 11-12, p. 1116–1121, 2007.

YAMAKOSHI, T. *et al.* Hemodynamic responses during simulated automobile driving in a monotonous situation. In: IEEE. **2006 International Conference of the IEEE Engineering in Medicine and Biology Society**. [S.l.], 2006. p. 5129–5132.

YANG, G.; LIN, Y.; BHATTACHARYA, P. A driver fatigue recognition model based on information fusion and dynamic bayesian network. **Information Sciences**, Elsevier, v. 180, n. 10, p. 1942–1954, 2010.

YU, C. *et al.* Drowsydet: A mobile application for real-time driver drowsiness detection. In: **2019 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCOM/I-OP/SCI)**. [S.l.: s.n.], 2019. p. 425–432.

YU, X. Real-time nonintrusive detection of driver drowsiness. University of Minnesota Center for Transportation Studies, 2009.