

# Sistema de Detecção de Fadiga Durante o Uso de um Veículo Automotivo

Antonio Prado da Silva Júnior  
Faculdade UNB Gama  
Universidade de Brasília  
Gama, Brasil  
Contato.pradojr@gmail.com

Ítalo Barbosa Santos  
Faculdade UNB Gama  
Universidade de Brasília  
Gama, Brasil  
Italo.b.s.35@gmail.com

**Resumo** — Neste documento está presente a proposta de um projeto para detectar fadiga de condutores utilizando uma raspberry pi e processamento de imagem a fim de reduzir o número de acidentes nas rodovias causados por sonolência do condutor.

**Keywords**—*Raspberry, processamento, imagem, câmera, estímulo.*

## I. JUSTIFICATIVA

A condução de um veículo durante o período da noite requer uma atenção maior do que nos outros períodos do dia. Entre os riscos mais comuns enfrentados pelos condutores está o cansaço e o sono provenientes de atividades realizadas durante o dia. Depois de 19 horas de privação de sono há diminuição de desempenho equivalente à observada em indivíduos com teor alcoólico no sangue de 0,70 g/l (aproximadamente igual a seis copos de cerveja ou três de vinho para um homem de 90 kg), segundo dados da Universidade Federal de São Paulo (Unifesp). De acordo com o ONSV (Observatório Nacional de Segurança Viária), dirigir com sono é equivalente a dirigir alcoolizado e a sonolência é responsável por até 50% dos acidentes em rodovias[1].

Para prevenir tais acidentes o uso de tecnologias embarcadas é indispensável. A empresa Ford possui um sistema de detecção de sonolência ou direção irregular implementada em alguns de seus carros, o nome deste sistema é Driver Alert. Sua funcionalidade se dá a uma câmera fixada nos retrovisores do carro que tem a finalidade de observar as faixas laterais da pista com isso há uma análise de como o veículo deveria reagir de acordo com o percurso, caso haja uma mudança brusca nessa previsão em comparação com a realidade, são emitidos alertas para o condutor. Conforme [2] é possível verificar a variação do batimento cardíaco do motorista e assim definir o nível de fadiga, porém o uso de

sensores para medir sinais vitais podem ser desconfortáveis para o usuário.

Com isso em mente o presente projeto busca evitar o contato direto com motorista utilizando imagens e técnicas de processamento de sinais para que o raspberry pi seja capaz de criar um alerta para o motorista se manter acordado ou tomar alguma medida necessária, desde parar no acostamento ou até ligar para algum familiar ou seguro veicular.

Para realizar o processamento de imagem é necessário determinar qual padrão deverá ser enfatizado, segundo[3] é possível detectar o nível de fadiga do motorista através do fechamentos dos olhos do condutor e de acordo com [4] é possível detectar o ato de bocejar do motorista, ambas formas de detecção representam níveis diferentes de fadiga. Logo o desafio é unificar ambos os meios de detecção a fim de criar alertas diferentes para cada nível de fadiga do condutor trazendo como benefício o aumento da segurança nas estradas durante o período noturno diminuindo a estatística de acidentes apresentados anteriormente.

## II. OBJETIVO

O objetivo deste projeto é desenvolver um sistema de monitoramento do motorista capaz de identificar fadiga por um meio não invasivo, a fim de, evitar acidentes graves com o motorista dormindo ao volante.

## III. REQUISITOS

Inicialmente para a realização do projeto podemos dividir os requisitos em Funcionais e Não Funcionais. Para os requisitos funcionais temos:

- O software deve reconhecer características faciais e ligá-las ao início da fadiga/sono;
- Ao detectar o sinal de fadiga/sono, emitir os estímulos para o condutor;

Enquanto para os requisitos não funcionais temos:

- Tempo de reconhecimento dos sinais de fadiga/sono e envio de estímulos para o condutor não deve ultrapassar os 5 seg;

O sistema de detecção de fadiga funcionará de tal forma a monitorar o comportamento facial do usuário por meio de uma câmera implantada dentro do veículo e direcionada ao rosto do condutor. Ao detectar tal padrão de fadiga o sistema enviará alertas sonoros e/ou físicos para estimular o condutor a manter-se acordado ou parar o veículo para tomar as medidas necessárias a fim de não provocar acidentes. Para a construção deste sistema serão utilizados os equipamentos da tabela 1 a seguir.

Quantidade	Material
01	Placa RaspBerry Pi3 Model B
01	Fonte 5v
01	Câmera
01	Led
01	Motor Dc
01	Buzzer
01	Suporte para acoplamento da case em carros
01	Case para Raspberry

Tabela 01 - Lista de equipamentos

A câmera será responsável pelo recebimento das imagens do rosto do condutor, o motor pela simulação do estímulo presente no banco do motorista, o buzzer para o estímulo sonoro depois que detectado o grau de fadiga/sono e a placa Raspberry Pi 3, que será responsável por todo o processamento.

#### IV. METODOLOGIA

##### A. Descrição do Hardware

A ligação dos componentes do projeto podem ser observados na figura 02 abaixo.

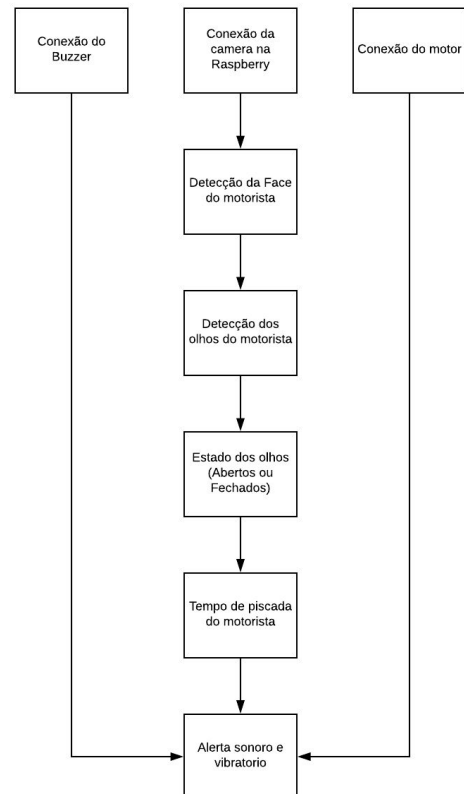


Figura 01 - Fluxo de desenvolvimento do projeto.

Na figura 1 está o fluxo de desenvolvimento do projeto que descreve os passos necessários para a conclusão do projeto. Como pode ser observado a primeira etapa consiste em conectar o hardware na raspberry pi 3 logo em seguida é feito a etapa do processamento de imagem sendo necessário primeiro detectar a face de uma pessoa e em seguida seus olhos. As duas etapas subsequentes são realizadas para verificar se o usuário de fato está cansado para isso será necessário verificar se os olhos do motorista está fechado por mais de dois frames seguidos e por fim é acionado um alerta sonoro e vibratório para manter o motorista acordado. A captura da imagem foi feita utilizando a biblioteca *Open Source Computer Vision (OpenCV)* com a linguagens de programação C e C++.

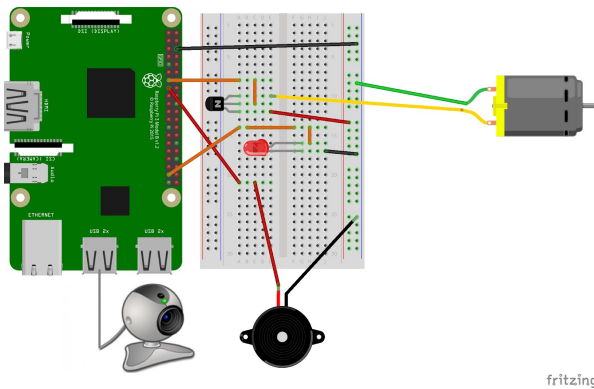


Figura 02 - Esquemático do projeto

A conexão do motor, buzzer e do led é basicamente com as portas GPIO da placa para os sinais e o GND para o aterramento, o motor é alimentado a partir do pino de 3.3V com o uso de um TBJ onde sua base está conectada no pino do VCC da placa, já o emissor e o coletor estão conectados nas duas outras pernas restantes do transistor, para a câmera a conexão é via usb da própria placa. A placa é alimentada por uma fonte de 5V e 3A.

Abaixo uma tabela descrevendo os preços de cada componente utilizado. O Kit da Raspberry Pi3 contém a placa com a fonte, case, dois dissipadores e um cartão de memória de 32GB.

Quantidade	Componente	Preço
1	Kit Raspberry Pi 3	R\$ 250,00
1	Câmera	R\$ 50,00
1	Kit 10 x Jumper	R\$ 7,50
1	Motor Dc	R\$ 5,00
1	Buzzer	R\$ 1,40
1	Led Vermelho	R\$ 0,25
1	TBJ BC549	R\$ 0,15
1	Resistor	R\$ 0,10
<b>TOTAL</b>		<b>R\$ 314,40</b>

Tabela 02 - Preço dos componentes do projeto

A estrutura do projeto é composta por duas partes, na primeira delas foi feito uma caixa de papelão de dimensões 10x7x3 cm. A segunda parte da estrutura corresponde a um suporte para encaixe de celular para carros.

Quantidade	Material	Preço
1x	Chapa de papelão	R\$ 2,00
1x	Suporte para carros	R\$ 15,00
<b>TOTAL</b>		<b>R\$ 17,00</b>

Tabela 03 - Preço dos materiais usados na estrutura

### B Dispositivos de alerta

Para a conexão dos dispositivos de alerta foi feito um código utilizando a biblioteca *wiringPi.h* que permite a utilização de função de escrita nos pinos de IO da raspberry com funções similares às presentes para Arduino como *pinMode* que permite definir o pino como entrada ou saída e *digitalWrite* a qual permite deixar o pino em nível lógico alto ou baixo ambas funções foram utilizadas para acionar o motor e o buzzer.

Como é necessário um alerta para sinalizar que o rosto está sendo detectado pela câmera um led foi adicionado que enquanto o rosto não estiver sendo identificado ele continuará piscando até que identifique.

Para o acionamento dos sinais em resposta ao estímulo proveniente do processamento foi criada uma thread que compreende tanto o acionamento do motor quanto o acionamento do buzzer.

### C. Detecção facial e dos olhos

O código para a conexão da câmera utilizou-se a classe *VideoCapture* a qual é utilizada para a captura de vídeo, imagem ou câmeras armazenando os dados obtidos em um variável do tipo *Mat* e utilizou-se as funções *.read* para ler os dados na variável a qual foi armazenada os dados da captura da câmera. Para realizar a detecção foram utilizados arquivos fornecidos pelo próprio *OpenCV* os *haarscascade*, esses arquivos possuem algoritmos previamente treinados utilizando o método de *AdaBoost* para detectar um cruciais de um determinado objeto[5], nesse caso foram utilizados os de detecção frontal e detecção dos olhos. O código consiste em carregar esses arquivos e logo em seguida realizar um tratamento simples de imagem fazendo a conversão da imagem para escala de cinza e por fim é feito uma equalização da distribuição dos tons de cinza da imagem. Depois desse tratamento é aplicado a *haarscascade* para detecção facial na matriz a qual contém a imagem capturada pela câmera, para executar essa ação foi necessário utilizar a função *detectMultiScale* colocando como parâmetros a matriz, o vetor que irá armazenar a imagem após o processamento, os valores mínimos dos pixels para ocorrer a detecção e os valores máximos, por fim foi desenhado uma elipse na face detectada para verificar o resultado obtido, porém os desenhos das elipses foram utilizados somente para testes, no código

final esse trecho foi retirado. O algoritmo da *haarscascade* foi aplicado no centro da imagem, primeiro detectando a face do usuário e com base na face detectada pela *haarscascade* extraiu-se a imagem que possuía somente a face e aplicou-se *haarscascade* para detectar os olhos do usuário.

A unificação entre os dispositivos de alerta e a detecção facial exigiu a utilização de um filtro. O filtro foi realizado utilizando um vetor de 10 posições com valores iniciais iguais a '0' em cada frame captado a posição lida do vetor era incrementada ao não identificar os olhos do usuário em determinada posição, seu valor era alterado de '0' para '1', caso o valor '1' fosse identificado em duas posições seguidas do vetor o programa ativa as threads dos alarmes as quais são mantidas ligadas até que não haja mais valores iguais a '1' seguidos, ao chegar na última posição do vetor é feito um vasculhamento de em todas as posições buscando valores diferentes de '0' caso encontrado algum o mesmo é atualizado para '0'.

## V. RESULTADOS

A conexão com a câmera foi bem sucedida sendo possível observar com clareza a imagem capturada com a câmera apesar da existência de um atraso entre a captura e a visualização na tela do computador. Os dispositivos de alerta foram testados primeiramente com um motor DC de porte médio, porém ao observar o risco de danificar os pinos de I/O da raspberry optou-se por trocar o motor por um de menor porte, pois não se sabia a corrente que o motor de médio exigia. Logo não era possível determinar qual transistor seria o adequado para o projeto. Os testes com os dispositivos de alarme foram feitos separadamente para verificar se era possível controlar o acionamento das thread de forma individual e cancelá-las quando necessário.

Nas figuras 3 e 4 está presente o resultado obtido para a detecção facial, porém esses resultados só são possíveis de se obter frontalmente e a uma distância correta, caso contrário os olhos do motorista são detectados mesmo ao serem fechados.

Ao unificar as partes dos dispositivos de alerta e da detecção facial os alarmes foram acionados com sucesso ao detectar os olhos do usuário fechados, porém notou-se um problema em relação a detecção indesejada do fechamento dos olhos quando o usuário pisca. Além de falsas detecções do rosto em paredes.

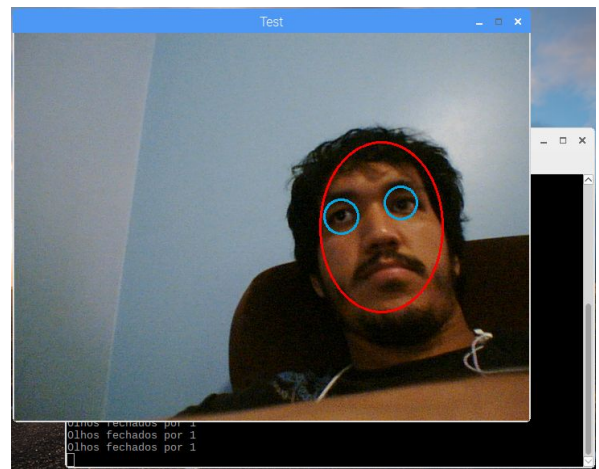


Figura 03 - Detecção facial com os olhos abertos.

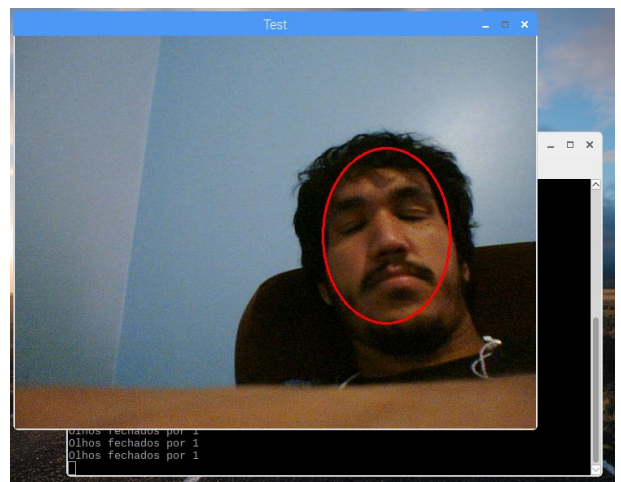


Figura 04 - Detecção facial com os olhos fechados

Consequentemente notou-se a necessidade de diminuir a taxa de captura das imagens para uma velocidade a qual o piscar dos olhos fossem de difícil detecção, para isso colocou-se uma taxa de 5fps, porém devido a programas do próprio sistema operacional e o tratamento da imagem realizado a taxa de amostragem se mostrou bem mais lenta que o desejado.

Mesmo com a taxa mais lenta o objetivo de evitar pegar a piscada do motorista foi alcançado, para realizar a validação da taxa de amostragem foram feitos 5 testes em frente a câmera com 20 piscadas em sequência, em dois testes os alarmes foram disparados, por isso viu-se a necessidade de aplicar um filtro para acionar os alarmes somente quando forem identificados os olhos fechados em dois frames seguidos. Após a aplicação do filtro os resultados desejados foram alcançados, mas é importante ressaltar que o sistema ainda possui falhas como a possibilidade de detectar uma piscada e um rosto em um local indesejado no frame seguinte essa é uma condição indesejada em que os alarmes serão acionados. Uma forma de contornar esse problema seria acionar os alarmes somente ao detectar em 3 frames seguidos,

porém não era uma opção viável, pois o sistema ficaria mais lento fazendo com que o requisito de acionar o alarme em menos de 5 segundos não fosse atendido.

Após realizar todos os testes com o auxílio da parte gráfica do sistema operacional do Raspbian, foram feitos novos testes sem a parte gráfica do projeto por meio da comunicação SSH.. Logo viu-se a necessidade de adicionar um LED para verificar se o rosto do usuário estava sendo detectado para testar essa parte foi necessário posicionar algum objeto na frente da câmera para observar se o LED piscava. Após realizar os mesmos testes piscando os olhos na frente da câmera deu-se início a reinstalação do sistema operacional Raspbian porém sem *Desktop*, a fim de acelerar o sistema, porém foi sem sucesso devido a um erro na instalação da biblioteca OpenCV como o tempo de entrega do projeto estava próximo optou-se por entregar o produto final em uma raspberry com *Desktop*.

A estrutura final do projeto conta com uma caixa onde estão presentes os sinais de alerta tal como o led de verificação da detecção do rosto. As imagens abaixo descrevem além da estrutura final, um dos testes feitos para a verificação da detecção facial inicial, na imagem à esquerda, o led ao lado do buzzer pisca intermitentemente até detectar um rosto a ser avaliado, já na imagem à direita é possível observar que o led está apagado sinalizando que existe um rosto detectado. Os demais testes realizados não foram registrados por se tratarem de um sinal sonoro e a vibração de um motor.



Figura 05 - Modelo final da estrutura

## VI. CONCLUSÕES

O projeto teve como objetivo principal desenvolver um método de detecção de fadiga no motorista que não fosse invasivo ao usuário, para isso utilizou-se uma câmera para detectar o momento em que o motorista dormiu ao volante. Logo foi necessário utilizar de técnicas de processamento de imagem para detectar a face e os olhos do usuário assim como dispositivos para gerar um alarme sonoro e vibratório.

Para o início do projeto o principal objetivo era realizar as conexões com a raspberry pi 3 e iniciar o processamento de

vídeo, porém devido a dificuldades durante a instalação da biblioteca openCV e a imprevistos nos dois dispositivos de alerta a única etapa concluída com sucesso do projeto foi a captura do vídeo da câmera. Após sanar esses problemas foi possível realizar o processamento da imagem e detectar a face e os olhos da pessoa tornando possível a unificação das partes, porém a detecção dos olhos fechados estavam sendo feitas quando o usuário piscava. Consequentemente acionando os alarmes em momentos indesejados, por isso viu-se a necessidade de verificar a taxa de amostragem do projeto assim como um filtro para acionar os alarmes somente se detectado o fechamento dos olhos em dois frames seguidos. Os testes realizados se mostraram satisfatórios, mas viu-se que era possível melhorar o sistema descartando a parte gráfica do sistema e utilizando um sistema operacional sem *Desktop*, porém foi sem sucesso e deixou-se o projeto na velocidade original. Ao finalizar o projeto como todo observou-se que o algoritmo utilizado nas *haarscascades* não são as ideias para essa aplicação, pois o algoritmo que detecta a face realiza falsas detecções em paredes ou em outros objetos enquanto ao algoritmo que detecta os olhos necessita de um posicionamento muito preciso para detectar o fechamento dos olhos, outros algoritmos mais complexos que realizam o desenho do contorno dos olhos por meio de pontos e verificam a distância entre esses pontos para determinar se houve ou não fechamento dos olhos seriam os ideais, mas por serem muito complexos foram inviáveis de se realizar no tempo estipulado para o projeto.

## REFERENCES

- [1] Sono e cansaço são responsáveis por até 50% dos acidentes em rodovias. Disponível em: <<https://www.onsv.org.br/sono-e-cansaco-sao-responsaveis-por-ate-50-dos-acidentes-em-rodovias/>>. Acesso em: 29 set. 2018.
- [2] E. Rogado, J.L. García, R. Barea, L.M. Bergasa, "Driver Fatigue Detection System", IEEE International Conference: Robotics and Biomimetics, 2008.
- [3] N. Alioua, A. Amine, M. Rziza1, "Driver's Fatigue Detection Based on Yawning Extraction", International Journal of Vehicular Technology, vol 2014.
- [4] D. L. Mena A.C.Rosero, "Portable Artificial Vision System to Determine Fatigue in a Person using a Raspberry PI3 card", IEEE International Conference on Information Systems and Computer Science, 2017
- [5] VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. In: IEEE. Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. [S.l.], 2001. v. 1, p. I-I.