

Detector de Sonolência e Distração no Trânsito

Guilherme dos Santos Araújo

*Programa de Graduação em Engenharia Eletrônica
Universidade de Brasília - FGA - Gama
Sistemas Operacionais Embarcados
E-mail: 160007658@aluno.unb.br*

Resumo—Neste ponto de controle será explicado como o projeto final da disciplina de *Sistemas Operacionais Embarcados*, proposto pela dupla, será construído ao longo do semestre.

I. INTRODUÇÃO

Segundo a Associação Brasileira de Medicina de Tráfego (ABRAMET), cerca de 74% dos acidentes de trânsito estão relacionados à falta de atenção do motorista, consequência de situações clínicas importantes (fadiga, stress, cansaço) e 8% à sonolência ao volante [1]. Considerado esse diagnóstico, o cansaço é uma das principais causas de morte nas rodovias brasileiras. Com o intuito de reduzir a ocorrência desses acidentes, o projeto visa indicar momentos de sonolência e distração do usuário, acionando alerta sonoro e luminoso para que ele saiba que está em uma situação perigosa e necessita mais atenção e de uma pausa para descanso até recompor sua energia para voltar a conduzir de forma segura.

A. Objetivos

Os objetivos que pretende-se alcançar são:

- Avisar ao usuário quando ele se encontra sonolento ou distraído ao volante por meio de alarme sonoro e luminoso.
- Reduzir acidentes de trânsito causados por sonolência e distração.
- Promover o bem estar e a segurança dos motoristas e da população.

B. Requisitos

Os requisitos para a construção física do projeto estão listadas a seguir:

- Uma placa Raspberry Pi 3 Model B V 1.2, mostrada na Figura 1.

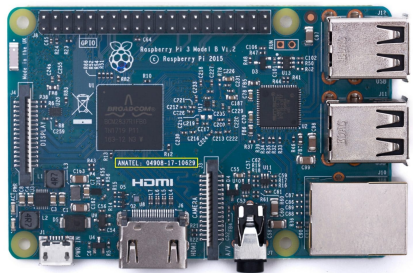


Figura 1. Raspberry Pi 3B V1.2.

Rafael Marinho Nunes

*Programa de Graduação em Engenharia Eletrônica
Universidade de Brasília - FGA - Gama
Sistemas Operacionais Embarcados
E-mail: 140159631@aluno.unb.br*

- Uma câmera Raspberry Pi Camera REV 1.3 para captar as imagens do rosto do motorista Figura 2.



Figura 2. Câmera Raspberry Pi Rev1.3.

- Sistema operacional: Raspberry Pi OS Figura 3.



Figura 3. Raspberry Pi OS.

- Case para o produto final (alocação do conjunto para monitoramento) Figura 4.



Figura 4. Case para câmera para fixação no automóvel.

C. Benefícios

Este projeto visa como benefício a redução de acidentes de trânsito com motoristas de caminhões e ônibus de viagens longas, melhorando também a segurança da população e motoristas de carros de passeio que transitam ou moram próximo às rotas, onde esse tipo de acidente por sonolência extrema ou distração do motorista acontecem.

II. METODOLOGIA

Neste trabalho, temos como objetivo criar um sistema de alerta sonoro em tempo real com base na observação visual do globo ocular do condutor, de modo a determinar a sua condição durante a condução do veículo. O sistema consiste em uma Raspberry pi 3, uma câmera usada para monitoração, um LEDs será inserido no pino 26, indicando quando o programa está em funcionamento, e um buzzer inserido no pino 27, que é acionado após detectar olhos cerrados durante alguns segundos.

A figura 5 indica como o sistema está configurado para detectar sonolência, a partir da proporção do olho do motorista e a figura 6 exemplifica como o sistema está montado.

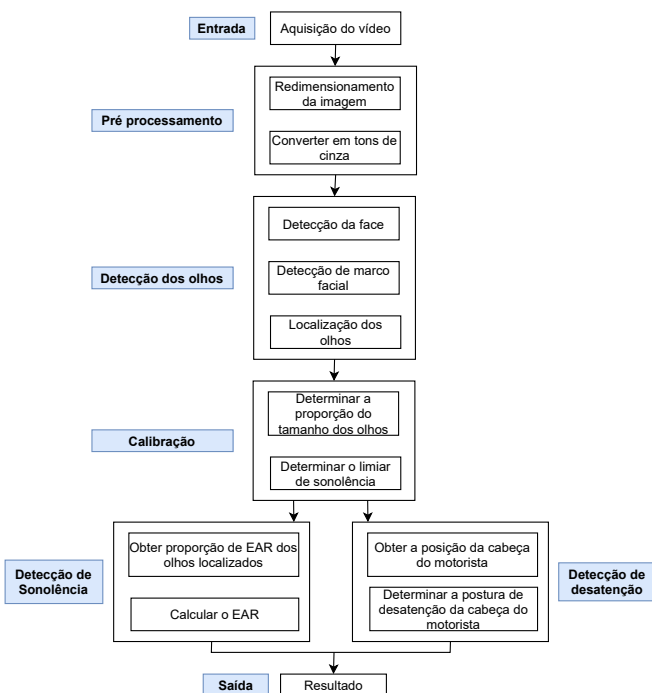


Figura 5. Fluxograma da detecção da face.



Figura 6. Componentes integrados com a raspberry.

A. Aquisição de video em tempo real

Uma câmera raspberry pi REV 1.3 será conectada na Raspberry pi 3 e posicionada em um case que estará fixo na frente do banco do motorista. A câmera deverá estar localizada de modo a capturar a face do condutor sem prejudicar a sua visão do trânsito. O sistema será energizado pela bateria do carro através de um cabo usb ou adaptador. Quando o sistema iniciar, a câmera vai capturar uma imagem em video HD do banco frontal do condutor. Em seguida, esse vídeo será usado a partir da captura de 30 frames por segundo codificado no formato H264.

B. Detecção de face

Nesta etapa, a imagem resultante é processada para obter 68 marcos faciais, usando a biblioteca de dados. Esta biblioteca detecta as características faciais da imagem de entrada e, em seguida, localiza todas as 68 coordenadas no rosto para determinar com precisão todas as características faciais, como sobrancelhas, nariz, boca, olhos e linha da mandíbula, como mostrado na figura 7. A região do olho é então usada para estimar com segurança a condição do olho [2].

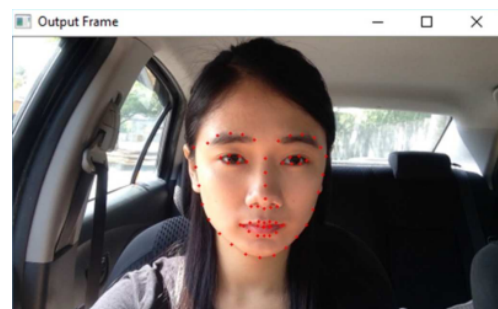


Figura 7. Detecção da face.

C. Detecção de olho

A localização da região do olho está sendo determinada a partir das 68 coordenadas do ponto de referência facial mapeadas, ou seja, o olho direito pode ser acessado através

dos pontos de referência 36 a 42, e o olho esquerdo pode ser acessado através dos pontos de referência 42 a 48 - veja a Figura 8. Os olhos localizados serão marcados usando a função de casco convexo disponível na biblioteca OpenCV. Mais tarde, a postura da cabeça é estimada usando a função solvePnP disponível na biblioteca OpenCV. As coordenadas globais calculadas, que são as coordenadas 3D de vários pontos de face, é usado para detectar a posição da cabeça. Esta pose é então reprojeta a partir dos pontos de coordenadas 3D para o eixo de coordenadas globais, mostrado na Figura 9.

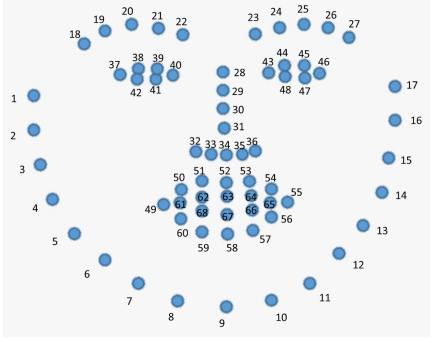


Figura 8. Mapeamento facial

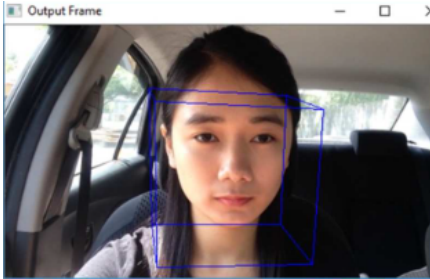


Figura 9. Detecção da cabeça

D. Cálculo do EAR

Nesta etapa, o EAR normal será determinado pelo sistema. Cada olho será detectado e representado por todas as seis coordenadas x y, para cada olho, como na Etapa C, mostrada na Figura 10, sendo os pontos P1, P2, P3, P4, P5 e P6. A equação EAR, mostrada na equação 1, calcula a distância entre os pontos de referência dos olhos verticais com a distância entre os pontos de referência dos olhos horizontais.

O EAR normalmente terá um valor constante quando o olho está aberto, enquanto cai bruscamente quando o olho está fechado. A Fig. 10 mostra a saída para EAR de cada quadro de uma câmera com 30 fps. Em experimentos preliminares, o EAR por 150 frames, coletados da fase de calibração de 5 segundos, será usado. Um valor de limite EAR, sendo este considerado como 0.25, foi definido. Um alarme só será acionado quando o EAR do motorista estiver abaixo do limite por um certo período de tempo, denominado limite de sonolência.

$$EAR = \frac{\|P2 - P6\| + \|P3 - P5\|}{P1 - P4} \quad (1)$$

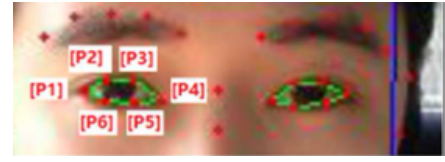


Figura 10. Calculando EAR

E. Detecção de sonolência

Qualquer quadro com EAR abaixo do limite de sonolência definido na Etapa 4 será registrado. Se para um número de quadros consecutivos, ou seja, por mais de 60 quadros, com EAR abaixo do limite de sonolência, um alarme será acionado para alertar o motorista, conforme mostrado na Figura 11. O número médio de quadros por segundo é de aproximadamente 30, e o número de quadros a serem considerados na detecção de sonolência (por 90 quadros) é igual a três segundos, o que significa que ao fechar levemente os olhos por 3 segundos, um alarme será acionado para alertar o motorista. O alarme será desligado assim que o motorista reabrir os olhos com um EAR acima do valor limite.

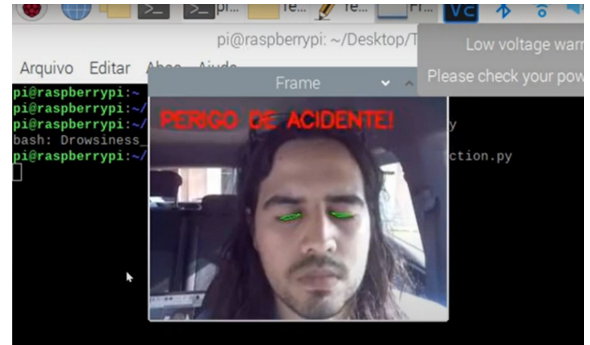


Figura 11. Captura do olho

III. RESULTADO EXPERIMENTAL

Foi realizado uma bateria de testes em 4 situações diferente para documentar a eficiência do sistema, encontrar falhas e defeitos e escolher o local mais adequado para fixação da câmera, como pode ser observado pela Figura 12.



Figura 12. Teste experimental

A. Teste 1: Sujeito 1 sem óculos de grau na sombra

No teste 1 foi feito a simulação em um cenário real, onde um condutor de um carro utiliza o circuito detector de sonolência no ambiente com sombra e sem óculos para verificar a aplicabilidade neste tipo de ambiente. O teste 1 obteve sucesso no reconhecimento da face de frente para a câmera e na detecção da sonolência no momento certo, onde os olhos estavam fechados por 10 segundos ou mais e o alerta foi acionado.

B. Teste 2: Sujeito 1 com óculos de grau na sombra

No teste 2 o condutor do carro continuava na sombra, porém agora o mesmo se encontrava de óculos. Nos primeiros teste nessas condições o circuito detector de sonolência apresentou falha no mapeamento do rosto, causado por uma má postura do condutor e também do ângulo diferente em que ele se encontrava. O teste 2 funcionou após o condutor ficar numa postura onde ele ficava de face para a câmera de frente, onde agora foi possível o mapeamento dos olhos e a detecção dos sinais de sonolência.

C. Teste 3: Sujeito 1 sem óculos de grau no sol

No teste 3 o ambiente foi alterado, condutor com veículo no sol e sem o uso de óculos. Foi notado que quando a luz do sol não iluminava diretamente no rosto ou na câmera e mantendo também a postura como nos casos anteriores, que é a postura ereta e face para câmera, a detecção da sonolência do condutor funcionou sem nenhum problema, alertando quando os olhos se mantinham fechados por 10 segundos ou mais.

D. Teste 4: Sujeito 1 com óculos de grau no sol

No teste 4, o condutor mudou o cenário na sombra para um ambiente ensolarado e utilizando óculos, onde foi notado que dependendo da intensidade da luz sobre o rosto e reflexo nos óculos, a detecção ficava um pouco imprecisa e sem o mapeamento dos olhos de forma correta. Com condições não extremas de luz solar diretamente sobre a face, óculos ou câmera e mantendo a postura corporal anterior, com a face frente à câmera, o teste sucedeu normalmente como nos testes anteriores e a detecção de sonolência foi realizada.

Número do Teste	Nº de Acionamentos Esperados	Nº de Acionamentos	Porcetagem de Acerto(%)
1	7	6	85,71
2	8	6	75
3	9	9	100
4	9	6	66,7

Figura 13. Resultado experimental dos testes

IV. CONCLUSÃO

Após a execução do código, obteve-se um bom percentual de aceitação (acima de 80%) na maioria dos casos, como exemplificado na Figura 13, o que significa um sucesso na criação e execução do programa. No caso do teste 4 (com óculos, em frente ao sol), ocorreu uma baixa porcentagem devido ao reflexo da lente frente ao sol forte, o que dificultou a captura do olho pelo Dlib.

Percebeu-se uma variação no nível de captura de frames da pi câmera, que fez com que o monitoramento do olho do condutor atrasasse em alguns segundos, cerca de 10 segundos de olhos cerrados para acionar a buzina, que foi reduzido para 4 segundos ao eliminar a exibição do vídeo em tempo real para a tela. Pode-se, então, dizer que a exibição em tempo real exige muito do processador da raspberry, que resulta na lentidão do sistema operacional.

A. Desenvolvimento Futuro

Possíveis melhoras são a criação de uma case mais compacta e gancho melhor adaptado para prender a raspberry e a câmera com o ângulo adequado. Outra necessidade é a otimização no software de modo a eliminar o delay de captura e aumentar a eficiência do alarme, e criação de uma função de stop acionada por um botão.

V. LINK DA APRESENTAÇÃO FINAL

<https://web.microsoftstream.com/video/46d3addc-a14c-424e-bec4-a8b9cf12ad00?list=studio>

REFERÊNCIAS

- [1] ABRAMET. *Problemas na saúde de motoristas causaram mais de 280 mil acidentes nas rodovias desde 2014*, aponta Abramet. Disponível em: <https://abramet.com.br/noticias/problemas-na-saude-de-motoristas-causaram-mais-de-280-mil-acidentes-nas-rodovias-desde-2014-aponta-abramet/>, Acesso em: 06 de Agosto de 2021.
- [2] Phooi Yee Lau Jie Yi Wong. *Real-Time Driver Alert System Using Raspberry Pi*. Disponível em: <https://ph02.tci-thaijo.org/index.php/ECTI-EEC/article/view/215488/149943>.