

## DETECÇÃO DE SONOLÊNCIA UTILIZANDO VISÃO COMPUTACIONAL

**Autores:** JOSÉ HÉLIO DE SOUZA, SILVIA RAMOS CRUZ

### Introdução

Atualmente um dos grandes fatores de ocorrência de acidentes em rodovias e no trabalho é o sono. Em uma pesquisa feita no Brasil, por Rizzo (2002), descreve que entre 1000 motoristas de todas as classes, neste 33% são caminhoneiros, dos quais 58% dirigem seus veículos com sono e 42% deles cochilavam ao dirigir. No total, 5% dos motoristas se envolveram em acidentes em consequência da sonolência. Dados de 2011 sobre o censo de 2008 do Ministério da Saúde mostram que o número geral de mortes por acidente de trânsito subiu para 38.273.

Com objetivo de reduzir os índices de acidentes e no trânsito e ou no trabalho, esse trabalho propõe o desenvolvimento de um sistema capaz de detectar sonolência utilizando uma câmera de baixa resolução e um computador capaz de processar e obter respostas em tempo real.

### Material e métodos

#### A. OpenCV

OpenCV (Open Source Computer Vision Library) é um programa gratuito onde tem um grande foco em aplicações em tempo real e eficiência computacional. Com interfaces C++, C, Python e Java e suporta Windows, Linux, Mac OS, iOS e Android.

#### B. Python

Python é uma linguagem de alto nível, orientada a objeto, funcional, imperativa, procedural, interpretada e de tipagem dinâmica. Devido a essas características exige poucas linhas de código, fácil leitura do código código se comparado o mesmo programa em outra linguagem. Ele prioriza legibilidade do código sobre a velocidade ou expressividade. Combina recursos de sua biblioteca padrão e por módulos e frameworks desenvolvidos por terceiros e a sintaxe concisa e clara.

#### C. Treinamento

Para realizar o treinamento do modelo classificador foi utilizado o programa cascade Trainer GUI. Onde foi utilizado o recurso LBP apesar de não ser tão preciso quanto o HAAR eles treinam muito mais rápido. As configurações de altura e largura da imagem foram 24 x 24, valores muito grandes tornam a detecção muito lenta.

#### D. Treinamento olhos fechados

No treinamento para detectar olhos fechados, foi necessário utilizar imagens positivas e negativas as nossas positivas são os olhos fechados sendo um total de 900 imagens, nas negativas foi inserida 500 imagens sendo elas de olhos abertos, rostos e outras quaisquer. Gerando o arquivo *xml haarcascade*.

#### E. Treinamento Bocejos

De maneira análoga foi feito o treinamento com bocejos. Porém com um banco de 355 imagens positivas e 251 imagens negativas, obtendo o arquivo *xml haarcascade*.

#### E. Detecção da Sonolência

Para resolver o problema de detecção de sonolência de motoristas e trabalhadores, dividimos-o em partes: a detecção da face, detecção dos olhos, classificação do estado dos olhos (abertos/fechados), detecção do bocejo, classificação do bocejo (bocejando/ não bocejando), identificação de sonolência.

Para verificação de olhos ou bocejos validados, foi realizado a contagem do número de detecções por um determinado tempo e de acordo com esse número, detectamos como sendo olhos fechados ou bocejos, evitando assim, não confundir com piscadas, rindo, conversando ou cantando.

De acordo com pesquisas o bocejo tem em média uma duração de 6,5 segundo, enquanto os olhos permanecem cerca de 3 segundos fechados para detectar uma sonolência. Levando em consideração esses dados, para detecção de sonolência foi usado o número de olhos fechados e bocejos em um determinado espaço de tempo.

## Resultados e discussão

### A. Teste Olhos fechados

Utilizando o arquivo *xml* de olhos fechados gerado pelo recurso do LBP, foi realizado os testes para validação, para focaliza e fazer a verificação somente na região do rosto utilizamos o arquivo *haarcascade\_frontalface\_alt.xml*, do pacote do OpenCv. Duas imagens geradas pela webCam podem ser vistas nas figuras 1 e 2, onde têm-se duas pessoas, onde pode ser visto a detecção dos olhos quando se fecha e a não detecção quando se encontra aberto.

### B. Teste Bocejos

Na validação do treinamento de bocejos, assim como na detecção de olhos fechados, foi utilizado a webcam e ao fazer gestos de bocejo o mesmo é identificado como pode ser visto nas figuras 3 e 4.

### B. Detectando Sonolência

Após os treinamentos e validação de olhos fechados e bocejos, foi realizada a junção dos dois, juntamente com a detecção de face, para se fazer uma busca mais precisa da região dos olhos e da boca. A figura 5 é uma imagem do vídeo teste rodando no programa desenvolvido, no momento em que ocorre a detecção de sonolência ao verificar olhos fechados.

## Considerações finais

Com a utilização do OpenCV a linguagem Python e o cascade Trainer GUI, foi elaborado um programa capaz de identificar a face, o estado dos olhos e bocejos, e a partir da identificação e análise da quantidade de detecções por tempo dos olhos fechados e ou bocejos emitir uma sinalização de sonolência. Contudo, chegamos a um resultado satisfatório com bons percentuais de detecção.

Para trabalhos futuros, propomos melhorias no programa, dentre elas no código e na detecção. Além de incluir novos sinais de sonolência, como: posições da mão no rosto e inclinação da cabeça. Deixando o programa ainda mais seguro e confiável.

## Referências bibliográficas

Rizzo, G. (2002). Drowsy driving in the south of brazil. *Revista brasileira Neurologia*, 68(3):94-96.

BERRI, Rafael Alceste et al. Detecção automática de sonolência em condutores de veículos utilizando imagens amplas e de baixa resolução. *Anais do Computer on the Beach*, p. 21-30, 2013.

<<https://www.python.org/>>. Acessado em: 10 set. 2017.

<<http://opencv.org/>>. Acessado em: 11 set. 2017.

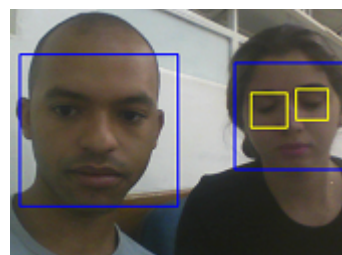
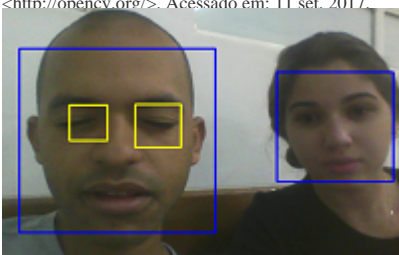




Figura 1. Detecção de olhos fechados



Figura 2. Detecção de olhos fechados

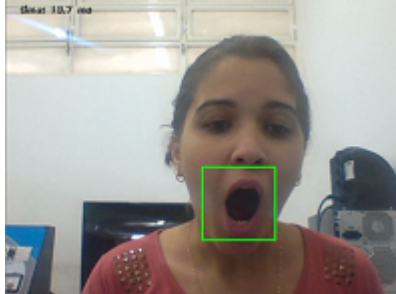


Figura 3. Detecção de bocejo

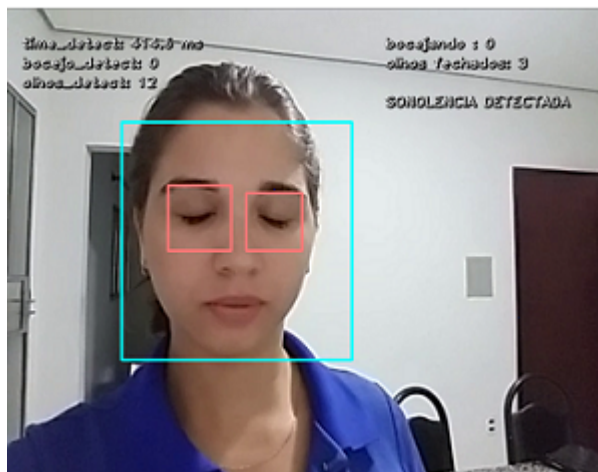
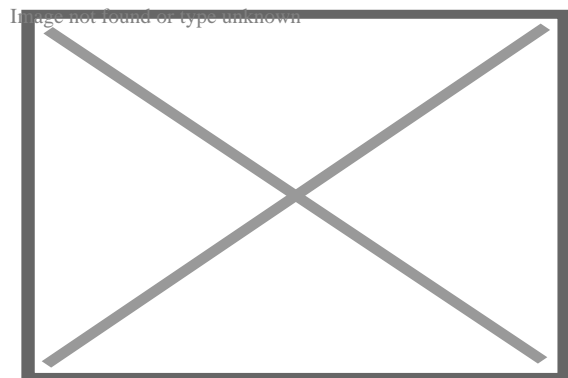


Figura 5. Detecção de Sonolência