



UNIVERSIDADE FEDERAL DO MARANHÃO  
Curso de Ciência da Computação

Danilo Nogueira Rodrigues  
**Detecção de Fadiga Baseada no  
Monitoramento dos Olhos**

São Luís - MA  
2018

Danilo Nogueira Rodrigues

# Detecção de Fadiga Baseada no Monitoramento dos Olhos

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, **como parte dos requisitos necessários** para obtenção do grau de Bacharel em Ciência da Computação.

Universidade Federal do Maranhão

Curso de Ciência da Computação

Orientador: Prof. Dr. João Dallyson Sousa de Almeida

São Luís - MA

2018

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).  
Núcleo Integrado de Bibliotecas/UFMA

Nogueira Rodrigues, Danilo.

Detecção de Fadiga Baseada no Monitoramento dos Olhos /  
Danilo Nogueira Rodrigues. - 2018.

33 f.

Orientador(a): João Dallyson Sousa de Almeida.

Monografia (Graduação) - Curso de Ciência da  
Computação, Universidade Federal do Maranhão, Universidade  
Federal do Maranhão, 2018.

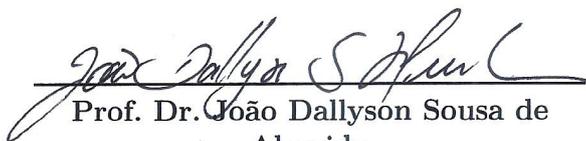
1. Detecção de Fadiga. 2. Detecção de Olhos. 3.  
Monitoramento. 4. Sistemas Inteligentes. 5. Visão  
Computacional. I. Sousa de Almeida, João Dallyson. II.  
Título.

Danilo Nogueira Rodrigues

## **Detecção de Fadiga Baseada no Monitoramento dos Olhos**

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, **como parte dos requisitos necessários** para obtenção do grau de Bacharel em Ciência da Computação.

Trabalho Aprovado em: São Luís - MA, 25 de Janeiro de 2018:



---

**Prof. Dr. João Dallyson Sousa de Almeida**

Orientador

Universidade Federal do Maranhão

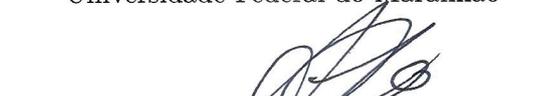


---

**Prof. Dr. Geraldo Braz Junior**

Examinador 1

Universidade Federal do Maranhão



---

**Prof. Dr. Anselmo Cardoso de Paiva**

Examinador 2

Universidade Federal do Maranhão

São Luís - MA

2018

# Agradecimentos

Ao meus pais, por me guiarem durante toda a minha educação e por conselhos que me ajudaram a escolher sempre o melhor caminho.

A minha irmã, Lays que nunca deixou de acreditar em mim e sempre me impulsionou em frente.

A minha família por sempre me apoiar e incentivar ao longo da minha jornada como estudante.

A minha namorada, Ananda que sempre esteve ao meu lado e me ajudou a alcançar meus objetivos.

Aos meus amigos, que por todos esses anos de curso sempre me ajudaram e motivaram.

Ao meus professores da UFMA, pela paciência e dedicação e todos os ensinamentos passados.

E a todos que contribuíram mesmo que indiretamente para o desenvolvimento deste trabalho.

# Resumo

O monitoramento de fadiga é um método automatizado de detectar o nível de cansaço de um indivíduo, com o intuito de melhorar a segurança em ambientes onde a atenção é um fator de risco. Nesta monografia é apresentado um método para detecção de fadiga em motoristas com a utilização de técnicas de processamento de imagem e aprendizado de máquina. Utiliza-se uma webcam para o monitoramento dos olhos em tempo real, considerando que a sonolência pode ser detectada com o tempo que o motorista permanece com os olhos fechados, a correta detecção do estado dos olhos (Abertos ou Fechados) é fundamental. Para a escolha da melhor técnica, dois algoritmos de aprendizado de máquina foram testados para a obtenção dos resultados e o desempenho comparado. Os resultados obtidos são medidos através da métrica de PERCLOS para ambientes controlados.

**Palavras-chaves:** Detecção de Olhos, Detecção de Fadiga, Monitoramento, Sistemas Inteligentes, Visão Computacional.

# Abstract

*Fatigue monitoring is an automated method of detecting an individual's drowsiness level with the intention of improve the security in a environment where attention is a crucial factor. In this work is introduced a method to fatigue detection in drivers using techniques of image processing and machine learning. A web cam is used to monitor eyes in real time, considering that somnolence can be detected with the time that a driver remains with eyes closed, the correct classification about eyes condition (Open or Closed) is fundamental. To choose the best technique, two machine learning algorithms was tested to obtain the results and thus the performance compared. The obtained results are measured with PERCLOS for controlled images*

**Keywords:** *Computer Vision, Eye Detection, Fatigue Detection, Smart Systems, Tracking.*

# Lista de abreviaturas e siglas

KNN	<i>K-Nearest Neighbors</i>
SVM	<i>Support Vector Machine</i>
HOG	<i>Histogram of oriented gradients</i>
ROI	<i>Region of interest</i>
CEW	<i>Closed Eyes In The Wild</i>
IBL	<i>Instance-Based Learning</i>
PERCLOS	<i>Percentage eye openness tracking</i>

# Sumário

	<b>Lista de ilustrações</b> . . . . .	<b>9</b>
<b>1</b>	<b>INTRODUÇÃO</b> . . . . .	<b>10</b>
<b>1.1</b>	<b>Objetivos</b> . . . . .	<b>11</b>
1.1.1	Geral . . . . .	11
1.1.2	Específicos . . . . .	11
<b>1.2</b>	<b>Organização do Trabalho</b> . . . . .	<b>11</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b> . . . . .	<b>13</b>
<b>2.1</b>	<b>Haar cascade</b> . . . . .	<b>13</b>
2.1.1	Características pseudo-haar . . . . .	13
2.1.2	Deteccção utilizando as características pseudo-haar. . . . .	14
<b>2.2</b>	<b>Histograma de gradientes orientados (HOG)</b> . . . . .	<b>16</b>
<b>2.3</b>	<b>K-nearest Neighbors</b> . . . . .	<b>18</b>
<b>2.4</b>	<b>SVM - Support Vector Machine</b> . . . . .	<b>19</b>
<b>3</b>	<b>METODOLOGIA PROPOSTA</b> . . . . .	<b>21</b>
<b>3.1</b>	<b>Extração do rosto e dos olhos</b> . . . . .	<b>22</b>
<b>3.2</b>	<b>Computação do Histograma de Gradientes (HOG)</b> . . . . .	<b>23</b>
<b>3.3</b>	<b>A base de dados</b> . . . . .	<b>24</b>
<b>3.4</b>	<b>Classificação</b> . . . . .	<b>25</b>
3.4.1	<i>K-Nearest neighbors</i> . . . . .	25
3.4.2	Máquina vetor de suporte . . . . .	25
3.4.3	Métricas para deteccção de fadiga . . . . .	25
<b>4</b>	<b>RESULTADOS E DISCUSSÃO</b> . . . . .	<b>27</b>
<b>4.1</b>	<b>Resultados dos testes dos classificadores</b> . . . . .	<b>27</b>
<b>4.2</b>	<b>Deteccção dos olhos</b> . . . . .	<b>28</b>
<b>4.3</b>	<b>Classificação do Nível de Fadiga</b> . . . . .	<b>29</b>
<b>5</b>	<b>CONCLUSÃO</b> . . . . .	<b>31</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>32</b>

## Lista de ilustrações

Figura 1 – Exemplo de características pseudo-haar comuns na detecção de rostos.	14
Figura 2 – Exemplos de características pseudo-haar de bordas, linhas e círculos.	15
Figura 3 – Processo de criação de uma imagem integral	16
Figura 4 – Kernels para a obtenção dos gradientes verticais e horizontais	16
Figura 5 – Procedimento de voto do HOG	17
Figura 6 – Exemplo das distancia das classes para o elemento no KNN	19
Figura 7 – Modelo de pontos no espaço do SVM	19
Figura 8 – Organização do método proposto	21
Figura 9 – Imagem RGB e <i>Grayscale</i>	22
Figura 10 – Exemplo de detecção de face e olhos com o <i>Haar Cascade</i>	23
Figura 11 – Visualização do histograma de gradientes de um olho	24
Figura 12 – Exemplos das imagens de olhos presentes na base de dados	25
Figura 13 – Comparação na acurácia para diferentes valores de K	28
Figura 14 – Visualização da classificação em tempo real	29

# 1 Introdução

O cansaço e a fadiga são fatores que aumentam bastante os riscos de acidentes no trânsito, normalmente a maioria dos motoristas não percebem ou imprudentemente não se importam que se encontram fadigados. Com o avanço de sistemas inteligentes, a detecção automática de fadiga pode ser feita com a análise de características físicas da face de uma pessoa.

Os olhos são uma das características mais relevantes para detecção de expressões, visto que quando uma pessoa se encontra em estado de sonolência, os olhos tendem a piscar mais do que o normal e com isso o tempo total com os olhos fechados é maior que o de uma pessoa em estado de descanso.

Utilizando esses dados visuais, seria possível criar um método automático de detecção inteligente. Esse método poderá medir a quantidade de fadiga que uma pessoa está sentindo, sendo treinado para detectar e classificar as imagens dos olhos e alertar o estado atual do motorista em tempo real.

Com isso esse método pode vir a ser útil para chamar a atenção do motorista e evitar um acidente, ou ainda alertar outros veículos de que o motorista não se encontra em seu estado pleno de atenção. Em casos de motoristas profissionais, o sistema poderia ser usado até para impedir o uso do veículo, evitando possíveis acidentes causados por fadiga antes mesmo de eles acontecerem.

A principal motivação para o desenvolvimento deste método é o problema grave de motoristas fadigados na direção. Esse problema é tão comum que todas as medidas possíveis devem ser tomadas a fim de evitá-lo. Com a tecnologia avançando cada vez mais rápido se tornou possível automatizar os sistemas de segurança de um veículo. Tal automatização pode literalmente salvar vidas, e é de interesse geral que o uso de métodos avançados para aumentar a segurança tanto do motorista, quanto a do trânsito como um todo.

Diversos trabalhos foram publicados ao longo dos anos sobre a detecção de fadiga do motorista [García et al. \(2010\)](#), [Silva et al. \(2012\)](#) e [Roque \(2013\)](#). É um tema bem consolidado e já vem sendo documentado por diversos pesquisadores. Em 2001 foi publicado um artigo falando sobre a correlação direta entre fadiga e acidentes [Fatigue \(2001\)](#). Essa correlação não é algo novo para a maioria dos motoristas, porém muitos imprudentemente dirigem em condições de fadiga.

Em 2010, um método para a detecção de fadiga em motoristas foi proposto [García et al. \(2010\)](#) onde foram utilizadas câmeras que capturam imagens em infravermelho

e técnicas de computação visual juntamente com o método PERCLOS para estimar o nível de fadiga, demonstrando que é possível se obter tal informação a partir apenas de informações visuais.

O trabalho de [Silva et al. \(2012\)](#) demonstrou diversas técnicas de visão computacional e aprendizado de máquina para detecção do nível de fadiga de um motorista. [Roque \(2013\)](#) demonstrou que é possível fazer esta tarefa apenas com técnicas de visão computacional, levando em consideração além dos olhos, regiões da boca e posicionamento da cabeça.

## 1.1 Objetivos

### 1.1.1 Geral

O objetivo geral deste trabalho é construir um método inteligente capaz de identificar fadiga em faces humanas em tempo real. A ideia é que a partir da detecção dos olhos, pode-se tirar uma conclusão sobre o estado de um motorista ao dirigir. Com a classificação dos olhos, pode-se prever se o olho se encontra aberto ou fechado e a quantidade de piscadas durante um determinado período de tempo e utilizar uma métrica para inferir o estado de fadiga do motorista.

### 1.1.2 Específicos

O sistema inteligente deve ser capaz de ser utilizado em tempo real, com algum dispositivo de captura portátil como por exemplo uma câmera. Deve ser possível a detecção a partir de um ponto fixo, mesmo em condições de luminosidade oscilantes e vibrações por conta dos movimentos do veículo.

Além destes objetivos, especificamente pretende-se:

- Contribuir para a construção de ferramentas automatizadas que possam ser aplicadas ao cotidiano da sociedade;
- Construir uma abordagem de funcionamento em tempo real;
- Contribuir para a construção de um método que possa reduzir estatísticas de acidente;

## 1.2 Organização do Trabalho

Este capítulo apresentou a introdução a motivação e os objetivos deste trabalho e listou os principais trabalhos relacionados.

O capítulo 2 trata de toda a fundamentação teórica aplicada neste trabalho. É fundamental um conhecimento básico sobre técnicas de processamento de imagem para o completo entendimento deste capítulo.

- A seção 2.1 e subseções apresenta a explicação do algoritmo responsável por realizar a detecção dos olhos criado por Viola e Jones ([VIOLA; JONES, 2001](#)).
- Na sub-seção 2.1.2 É explicado a forma de detecção do algoritmo *haar cascade*.
- A Seção 2.2 apresenta a técnica de extração de características de imagem utilizada neste trabalho chamada HOG.
- A Seção 2.3 fundamenta a teoria básica do classificador KNN.
- A seção 2.4 o funcionamento de outro classificador é explicado, a Máquina Vetor de Suporte (SVM).

No capítulo 3 a metodologia é proposta. Esse capítulo apresenta a metodologia desenvolvida a partir da aplicação das técnicas descritas no Capítulo 2. A metodologia é composta por um conjunto de etapas, cada uma descrita em uma das seções deste capítulo.

- Na seção 3.1 todo o processo de extração do rosto e dos olhos é detalhado.
- Na 3.2 é demonstrado o processo realizado para a extração de características usando o HOG.
- Em 3.3 a base de dados é introduzida e explicada.
- Em 3.4 O processo de classificação é introduzido
- Na subseção 3.4.1 o processo de classificação usando KNN é demonstrado.
- Na subseção 3.4.2 o processo de classificação usando SVM é demonstrado.
- Na subseção 3.4.3 é introduzido o método para a detecção de fadiga o PERCLOS.

No capítulo 4 são apresentados os resultados de testes, as métricas de desempenho e a discussão dos resultados obtidos.

- No item 4.1 os classificadores são comparados, utilizando a base de dados CEW.
- No item 4.2 o método de detecção dos olhos é avaliado e comentado.
- O item 4.3 descreve a classificação dos olhos e estimativa de fadiga utilizando o PERCLOS.

No capítulo 5 são apresentadas as considerações finais da metodologia implementada, comentários sobre problemas e conclusões notadas no termino do trabalho.

## 2 Fundamentação teórica

Com o crescente aumento da quantidade de veículos no mundo todo, é de se esperar que as pessoas passem a utilizar os veículos automotores como método principal de transporte. Hoje, a quantidade de carros cresce mais do que a população mundial (NYT, 1997) e com isso, cresce também o total de acidentes. Vários estudos sugerem que cerca de 20% de todos os acidentes de carros são relacionados a fadiga, e até 50% em algumas estradas (FATIGUE, 2001) e (COMPASS, 2015).

A utilização de veículos para uso comercial também segue o mesmo ritmo. Caminhoneiros, motoristas de táxi, Uber ou ônibus costumam dirigir por muitas horas seguidas, utilizando-se até remédios e energéticos para inibir o sono e o cansaço físico e mental. Tal prática é considerada extremamente arriscada, de acordo com a pesquisas recentes, dirigir em estado de fadiga pode ser mais arriscado do que alcoolizado (CBS, 2016).

Felizmente, com o avanço tecnológico é possível criar um sistema inteligente para monitorar o motorista em tempo real, evitando um possível acidente. No entanto, esta não é uma tarefa simples. Existem diversas técnicas de detecção e classificação de objetos em imagens, porém a maioria não é rápida o suficiente para uma detecção em tempo real. Assim como a detecção, a classificação também é difícil, ainda mais quando o conjunto de dados a ser classificado não se encontra em condições controladas, o que pode gerar erros de classificação.

### 2.1 Haar cascade

Dentre as diversas técnicas que já se encontram bem fundamentadas na literatura, a detecção de objetos utilizando o algoritmo *Haar Cascade* é uma das mais rápidas, sendo bastante utilizada em diversas aplicações de computação visual para detecção de faces e outros objetos.

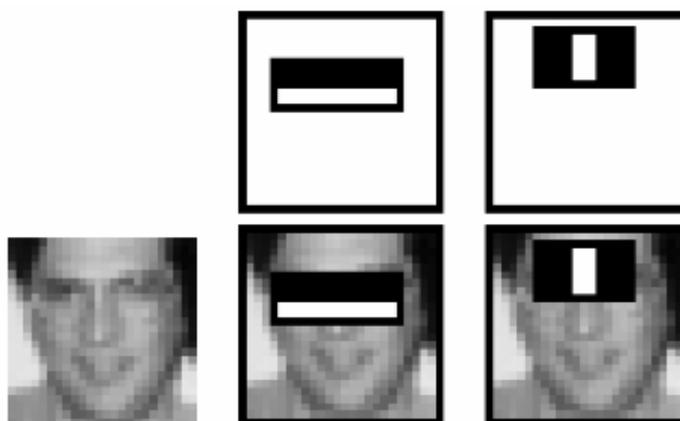
#### 2.1.1 Características pseudo-haar

Uma *haar-like feature* é uma característica na forma de um pequeno padrão de pixels que são utilizadas para detecção de objetos, como um kernel simples. Inicialmente o uso de todos os valores de pixels presentes em imagens tornava o trabalho de extração de características extremamente custoso em um ponto de vista computacional. Buscando meios de sobrepujar o problema do custo computacional, Papageorgiou et al (PAPAGEORGIOU; OREN; POGGIO, 1998) propuseram o uso de uma nova forma de extração de características com um custo computacional menor, se comparado com o método tradicional. A alternativa

proposta foi da utilização de um método baseado nas Haar wavelets ou "Transformada de Haar" que é um tipo de transformada matemática discreta usada no processamento e análise de sinais.

A adaptação dessa ideia foi feita por Viola e Jones (VIOLA; JONES, 2001), que usaram as *haar wavelets* para criar as chamadas características pseudo-haar. Uma *haar-like feature* funciona levando em consideração as regiões em volta de uma determinada coordenada em uma janela de detecção em uma imagem, e soma todas as respectivas intensidades de pixels em cada uma dessas coordenadas e em seguida calcula a diferença entre essas somas. Posteriormente, estas diferenças são utilizadas para categorizar a imagem em subseções que agem como fronteiras limitadoras para o objeto que se deseja detectar. Um exemplo comum é a detecção de rostos humanos: uma característica comum entre todos os rostos é que a região onde ficam os olhos é bem mais escura do que se comparadas com o resto do rosto.

Figura 1 – Exemplo de características pseudo-haar comuns na detecção de rostos.



Fonte: (VIOLA; JONES, 2001)

Logo, a escolha de uma característica haar que seja um par de retângulos adjacentes que ficam por cima do olho e da região das bochechas é considerada boa para a detecção de rostos. A posição desses retângulos é obtida com base na janela de detecção que foi encontrada no passo anterior e age como espaço delimitador para a detecção do objeto em questão.

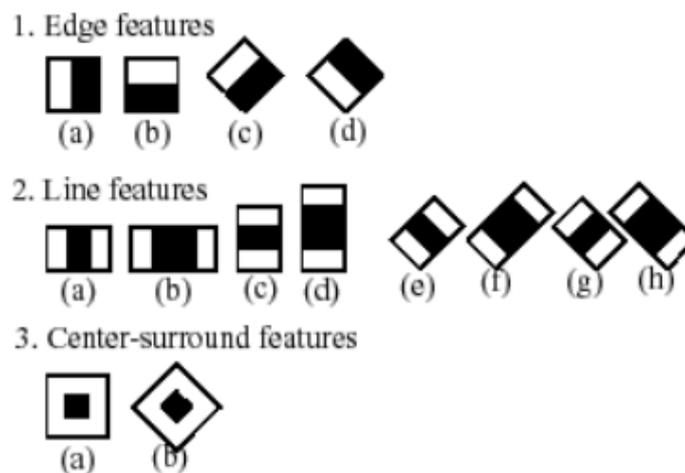
### 2.1.2 Detecção utilizando as características pseudo-haar.

O processo de detecção proposto por Viola e Jones (2001), consiste em uma "janela de detecção" com tamanho aproximado do objeto do qual será detectado, que "desliza" pela imagem (como em uma convolução) e para cada subseção da imagem, várias características pseudo-haar são calculadas. As diferenças são então comparadas com um limiar preestabelecido que separa possíveis objetos de todo o resto. Como uma só

característica pseudo-haar é extremamente simples, ela por si só não tem confiabilidade suficiente para descrever com precisão o objeto a ser detectado, é necessário um número bem grande de características para isso. No classificador proposto por Viola e Jones essas características são agrupadas em um formato conhecido como "classificação em cascata".

A palavra "cascata" nesse contexto, significa que a classificação resultante consiste de alguns classificadores bem simples que agem como estágios de classificação, sendo aplicados de maneira subsequente em uma região de interesse, até que em algum estágio o candidato seja rejeitado ou aceito em todas as etapas. Os classificadores usados em cada estágio da "cascata", possuem por si só uma certa complexidade. Eles são constituídos com base em classificadores básicos como árvores de decisão de pelo menos duas folhas e ao mesmo tempo usando um algoritmo de melhora de performance. O *Adaboost* foi o algoritmo utilizado neste trabalho, e ele utiliza um sistema de votos com pesos para aumentar a eficiência.

Figura 2 – Exemplos de características pseudo-haar de bordas, linhas e círculos.

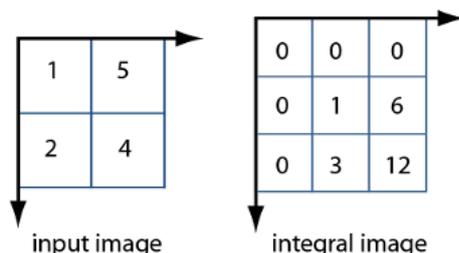


Fonte: (VIOLA; JONES, 2001)

A grande vantagem das características do tipo haar é a performance. Por serem imagens "integrais" elas podem ser facilmente calculadas em um tempo curto e constante pelo processador, o que torna esse método de detecção de objetos ideal para uma detecção em tempo real.

Imagens integrais são tabelas de consulta onde os valores de cada célula é resultante das somas das células acima e a esquerda, logo é possível saber a soma de qualquer retângulo dentro da imagem apenas com quatro (ou mais) coordenadas, ao invés de somar até chegar ao determinado ponto.

Figura 3 – Processo de criação de uma imagem integral



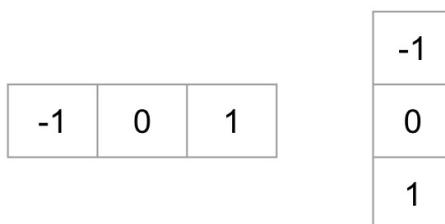
Fonte: Acervo do autor

## 2.2 Histograma de gradientes orientados (HOG)

O histograma de gradientes orientados (HOG) é o descritor de características sendo usado com foco principalmente na detecção de objetos. O HOG é considerado um descritor denso, nesse contexto significando que ele extrai características de todas as regiões de uma imagem, ou de alguma área de interesse. O uso do HOG foi popularizado por Dalal e Triggs (2005) no trabalho sobre detecção de pedestres. Em 2006 eles também estenderam o algoritmo e publicaram a aplicação de HOG para objetos comuns (DALAL; TRIGGS, 2006) provando a eficácia do método.

O processo para conseguir o histograma de gradientes orientados se inicia no cálculo dos gradientes (mudanças direcionais na intensidade dos pixels) de uma dada imagem. Para isso são aplicados dois filtros por toda a imagem para se obter os gradientes verticais e horizontais.

Figura 4 – Kernels para a obtenção dos gradientes verticais e horizontais



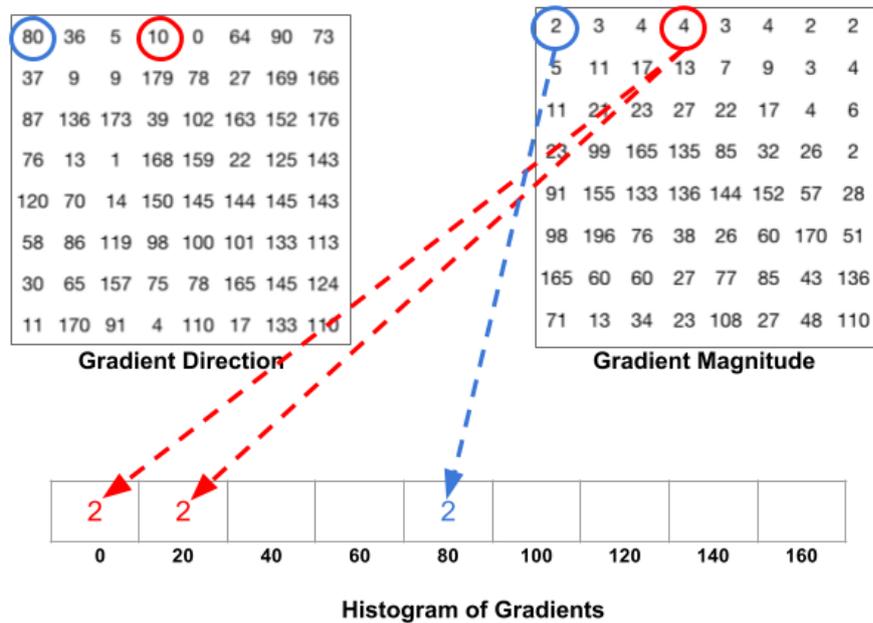
Fonte: Acervo do autor

O descritor tenta capturar informações sobre a forma de estruturas na imagem utilizando informações sobre os gradientes. O HOG extrai estas informações dividindo a imagem em quadrantes menores de 8x8 e subdividindo estes em blocos de 3x3 células, aonde cada célula possui um número fixo de orientações com um voto proporcional a magnitude do pixel.

Estes quadrantes de 8x8 são uma escolha do algoritmo em si, o valor deve ser

ajustado de acordo com o objeto no qual a extração de características está sendo aplicada. O histograma de gradientes é essencialmente um vetor de 9 células correspondendo aos ângulos  $0^\circ$ ,  $20^\circ$ ,  $40^\circ$ ,  $60^\circ$  ...  $160^\circ$ . O procedimento de criação do vetor de gradientes é ilustrado na Figura 5. Os valores de magnitude e direção de cada quadrante  $8 \times 8$  é selecionado com base na direção e o valor a ser colocado no vetor de gradientes é selecionado com base na magnitude. Caso o ângulo seja maior que  $160^\circ$  ele se envolve em si mesmo, logo  $0^\circ$  e  $160^\circ$  passam a ser equivalentes fazendo com que nesse caso, ambos os ângulos recebem um voto proporcional.

Figura 5 – Procedimento de voto do HOG



Fonte: Acervo do autor

Por fim, é obtido um vetor contendo o histograma dos ângulos dos gradientes, porém gradientes são sensíveis a luz, fazendo com que dependendo da iluminação os valores de todos os gradientes mudem drasticamente. Para iniciar o processo de normalização de uma imagem RGB (3 canais de cor) é utilizada a norma L2 (Equação 2.1) que consiste na operação mostrada na equação abaixo. Assim que calculada a norma ela é usada para dividir os valores dos pixels por toda a imagem normalizando os valores. O próximo passo é calcular todo o vetor HOG de características, para isso basta concatenar todos os vetores obtidos de cada quadrante, finalmente obtendo o vetor final de características.

$$f = \frac{\vartheta}{\sqrt{\|\vartheta\|_2^2 + e^2}} \tag{2.1}$$

## 2.3 K-nearest Neighbors

O Algoritmo K-nearest Neighbors (KNN) é um método de reconhecimento de padrões usado para classificação ou regressão (ALTMAN, 1992). Na classificação, o objetivo é encontrar um número pré-definido de amostras de treino mais próximas do novo ponto e medir a distância ente eles, e com base nesta distância, prever a classe a qual este ponto pertence.

Apesar da simplicidade, o KNN é bem-sucedido em uma grande quantidade de problemas de classificação. Por ser um método não paramétrico, é comumente bem-sucedido em situações onde a situação das fronteiras de classificação é bem irregular.

O teinamento do KNN é feito com vetores em um espaço multidimensional, cada um com uma classe associada. Esses vetores são então armazenados para a fase de classificação. Na fase de classificação o valor de K é definido pelo usuário, e um vetor a ser identificado recebendo a classe na qual é mais frequentemente próximo.

A distância é medida geralmente utilizando a distância euclidiana (Equação 2.2), que é adequada para valores contínuos. Um problema comum do KNN é que quando alguma classe é mais comum do que outras, ela tende a ser sempre a maioria dos votos. O que pode acarretar em classificações incorretas. Uma solução é utilizar pesos que são inversamente proporcionais a distância.

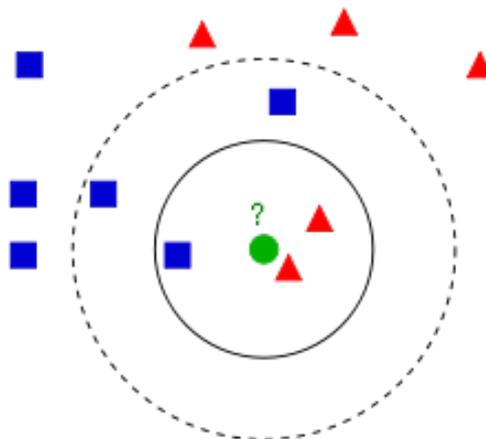
$$\sqrt{\sum_{i=1}^k (x_i - y_i)^2} \quad (2.2)$$

A escolha do valor de K geralmente depende do tipo de dado a ser classificado. Dados binários (apenas duas classes) geralmente são ideais com valores de K ímpares, a fim de evitar empates. Valores altos para K geram uma classificação mais resistente a ruído, no entanto a fronteira entre as classes acaba ficando menos distintas. A performance do KNN é severamente afetada se o conjunto de dados usado é extremamente ruidoso, é necessário utilizar de processos de normalização e extração de características relevantes nos dados para a efetividade da classificação melhorar.

O KNN é um tipo de classificador da família IBL (*Instance-Based Learning*) que significa que ele é um tipo de algoritmo onde ao invés de fazer uma generalização explícita, ele compara o elemento a ser classificado com instâncias vistas no treino, que são armazenadas em memória. A desvantagem disso é que a classificação da hipótese cresce de acordo com a quantidade de dados do treinamento (RUSSELL; NORVIG; INTELLIGENCE, 1995).

Dessa forma, o KNN é uma maneira simples de averiguar o funcionamento do sistema, levando em consideração a quantidade de dados do *dataset* e a natureza do problema de classificação.

Figura 6 – Exemplo das distancia das classes para o elemento no KNN



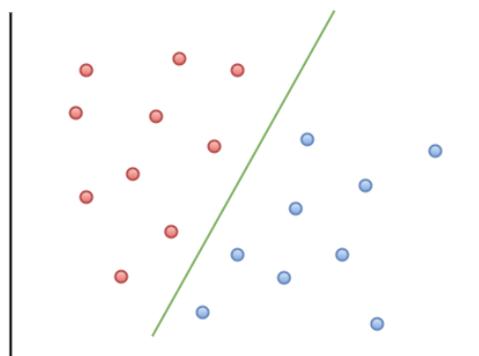
Fonte: Acervo do autor

## 2.4 SVM - Support Vector Machine

Support Vector Machine (SVM) ou máquina de vetores de suporte, é um método de aprendizado supervisionado associado com algoritmos de aprendizagem por classificação ou regressão. Foi originalmente proposto por [Cortes e Vapnik \(1995\)](#) e publicado em 1995 .

De maneira simples, um SVM prediz para um dado grupo de dados a qual das duas classe existentes a entrada fornecida pertence, o que faz do SVM um classificador linear binário não probabilístico. Para obter a predição, é necessário primeiro fornecer um conjunto de treino, contendo exemplos rotulados para cada classe. Com estes dados o algoritmo do SVM produz um modelo que atribui novos exemplos a uma categoria ou outra.

Figura 7 – Modelo de pontos no espaço do SVM



Fonte: Acervo do autor

O modelo criado pelo SVM é uma demonstração dos pontos no espaço colocados de forma a que os exemplos de cada classe fiquem separados por uma máxima distância possível, como exemplificado na Figura 7. Este modelo é o mesmo usado para definir a classe de um novo exemplo, colocando-o em um ponto do modelo e determinando de o

mesmo pertence a uma classe ou outra. Essa linha de separação encontrada pelo SVM entre as classes é chamada de hiperplano. A margem do hiperplano tende a ser expandida ao máximo pelo algoritmo durante o treinamento até que por fim, as margens sejam definidas.

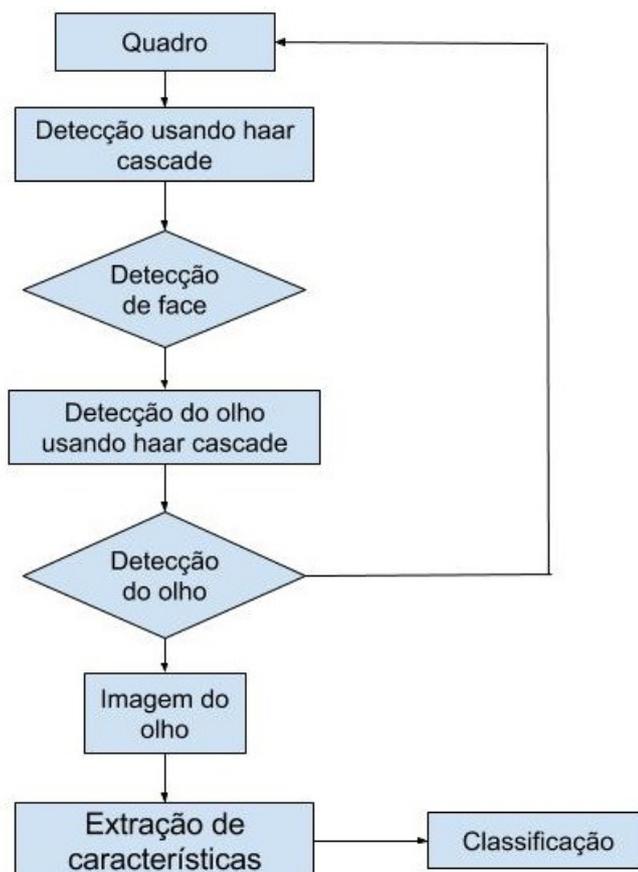
O algoritmo do SVM é um classificador criado com o intuito de classificar dados linearmente separáveis, isso significa que o classificador não lida bem com dados ruidosos, ou ainda classes muito semelhantes. O SVM tenta encontrar a melhor maneira possível de classificar os dados, em conjuntos de dados que possuem anomalias (pontos no modelo que estão totalmente separados de sua classe) podem vir a ser desconsiderados a fim de melhorar a classificação.

### 3 Metodologia Proposta

A metodologia proposta para este trabalho tem como objetivo principal a detecção e classificação do estado dos olhos de uma pessoa, utilizando imagens provenientes de uma câmara em tempo real e a associação destes estados do olho (aberto ou fechado) ao longo do tempo, com um nível de fadiga da pessoa observada. O procedimento é composto nas seguintes etapas: as imagens digitais são capturadas utilizando uma câmara em tempo real (30 quadros por segundo) e a partir de cada quadro, a face é detectada utilizando o algoritmo *haar cascade*, em seguida o mesmo algoritmo também detecta os olhos dentro da região da face previamente detectada.

Com os dados da região dos olhos, é possível iniciar o processo de extração de características utilizando o Histograma de Gradientes Orientados (HOG). Os valores obtidos pelo HOG são então usados na etapa de classificação onde o KNN ou SVM são utilizados para determinar a que classe a imagem pertence.

Figura 8 – Organização do método proposto



Fonte: Acervo do autor

Para detectar o estado de sonolência de uma pessoa, foram escolhidas duas métricas:

A métrica de PERCLOS (Calcula a proporção de tempo em que os olhos se encontram fechados) e o ritmo de piscadas do indivíduo.

### 3.1 Extração do rosto e dos olhos

A extração do rosto e dos olhos segue um conjunto de etapas para a obtenção da ROI (região de interesse), cada etapa realizando uma ou mais funções específicas:

- Captura de imagem.
- Tratamento da imagem capturada.
- Detecção do rosto.
- Detecção dos olhos.
- Ajustes na imagem após a detecção.

Inicialmente a captura da imagem é feita utilizando uma câmera digital simples, como uma *webcam*. No problema proposto neste trabalho, a câmera ficaria posicionada dentro do veículo em uma posição fixa e em um ângulo no qual a posição do rosto esteja visível frontalmente. Por conta de ser um ambiente parcialmente isolado (dentro do veículo) problemas como, oclusão parcial, ou total, escala e deformação são praticamente inexistentes. No entanto, variações de luz ou a vibração do veículo em si, podem acarretar em problemas na detecção.

A câmera produz continuamente um fluxo de 30 quadros por segundo, tornando o fluxo de dados bem grande. Para isso é aplicada uma transformação nos canais de cores de vermelho, verde e azul (RGB) para escala de cinza, dessa forma a quantidade de pixels da imagem capturada é reduzida em três vezes.

Figura 9 – Imagem RGB e *Grayscale*

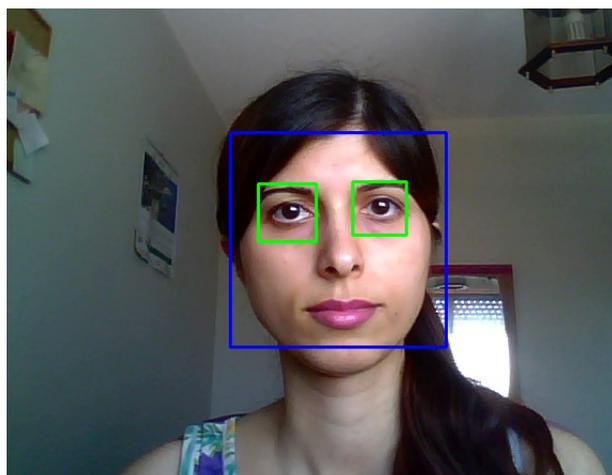


Fonte: Acervo do autor

Em seguida o *Haar cascade* é utilizado para localizar o rosto dentro da imagem. O algoritmo é pré-treinado para detectar rostos e olhos, a detecção do rosto é importante nesse caso, pois irá definir a ROI onde o *haar cascade* novamente irá detectar os olhos. Como os olhos são geralmente pontos pequenos dentro da imagem, o *haar cascade* pode perder muito tempo procurando no pior caso, onde olhos estejam próximos do local final da varredura. A face, por ser consideravelmente maior é mais fácil de detectar e por isso torna a área varredura consideravelmente menor do que toda a imagem.

Assim que o *Haar cascade* detecta um olho, ele retorna as coordenadas dentro do quadro da posição do olho. Dado a natureza do problema, por conta de vibrações ou até da própria detecção do *Haar cascade* as coordenadas podem incluir parte ou toda a sobrancelha do indivíduo, causando problemas na etapa de classificação. Para contornar este problema o tamanho das imagens foi padronizado e a região da sobrancelha é automaticamente cortada da área de detecção.

Figura 10 – Exemplo de detecção de face e olhos com o *Haar Cascade*



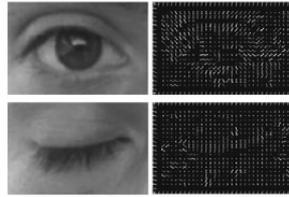
Fonte: Acervo do autor

## 3.2 Computação do Histograma de Gradientes (HOG)

Na próxima etapa é extraída a característica que vai ser usada para a classificação, o HOG. Este descritor foi especificamente escolhido pelo fato de sua ideia principal ser que a aparência e a forma dos objetos podem ser descritas através da distribuição da intensidade de pixels. Como o olho possui uma forma bem distinta de outras características faciais, tanto aberto quanto fechado, o HOG é capaz de extrair essas características de maneira extremamente rápida e detalhada.

Um dos problemas enfrentados nas imagens é a variância de cor e sombras, muito comum em um ambiente como a parte interna de um veículo. O uso do HOG se justifica também pelo fato de ele utilizar um processo de normalização na imagem, que resulta numa

Figura 11 – Visualização do histograma de gradientes de um olho



Fonte: Acervo do autor

melhor invariância de cor e sombras na imagem, facilitando o processo de classificação. Um fator determinante para o uso desta técnica é que o HOG é muito bom com imagens pequenas, o que é o caso pois em parte do algoritmo do HOG as operações passam ter um custo computacional elevado para imagens em alta resolução.

Para a caracterização das imagens, o HOG foi configurado com a dimensão 24x24, com os seguintes parâmetros: o tamanho de bloco igual a 4x4 e com células de tamanho 2x2. Como a imagem obtida é muito pequena, células maiores de 2x2 pixels não teriam resultados satisfatórios e características específicas dos olhos poderiam não ser selecionadas.

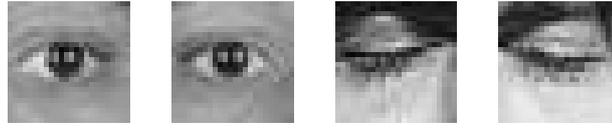
### 3.3 A base de dados

A escolha da base de dados para este trabalho levou em consideração diversos fatores: a diversidade de indivíduos, a qualidade dos dados, a quantidade total de imagens e o posicionamento das imagens. Uma base de dados muito pequena poderia facilmente ter problemas durante o treino, visto que o classificador precisa de alguns exemplos para um treino satisfatório. O posicionamento também é importante, já que foi predefinido que neste trabalho a detecção seria realizada frontalmente, devido o posicionamento da câmera. Quanto a diversidade, a base de dados em questão possui 2423 indivíduos diferentes o que certamente é suficiente para solucionar o problema.

Neste trabalho foi utilizada a base de imagens *Closed Eyes In The Wild (CEW)* (SONG et al., 2014). Esta base possui 4809 imagens de olhos, divididas em 4 grupos para cada tipo de olho (olho esquerdo aberto, olho direito aberto, olho da esquerdo fechado e olho direito fechado). As imagens tem resolução padronizada em 24x24 pixels e já se encontram em escala de cinza (Figura 12).

Para a solução do problema, a classificação diferenciada do olho esquerdo ou direito não era importante, portanto, as classes foram espelhadas e somente duas classes foram utilizadas para classificação (Aberto ou Fechado).

Figura 12 – Exemplos das imagens de olhos presentes na base de dados



Fonte: (SONG et al., 2014)

## 3.4 Classificação

O processo de classificação foi feito pensando nos tipos de dados que seriam utilizados. Imagens adquiridas naturalmente a partir de uma câmera podem conter muitas variações de luz e sombra, sendo necessário realizar algum tipo de tratamento. Porém o método de extração de características ajudou bastante nesse quesito, normalizando os dados e tornando-os aptos para fornecer ao classificador.

Para a classificação dos olhos, foram escolhidos dois algoritmos: a Máquina Vetor de Suporte (SVM) e o *K-Nearest Neighbors* (KNN). Para a construção do sistema em python, uma biblioteca de data-mining e análise de dados foi utilizada, o scikit-learn.

### 3.4.1 *K-Nearest neighbors*

O KNN é um algoritmo que está entre os mais simples na área de aprendizado de máquina e é comumente usado em classificação. O mesmo foi configurado com distância euclidiana. A quantidade de  $K$  indica o numero de votos em que o algoritmo irá se basear para efetuar a classificação. No caso do problema em questão, foram testados três valores para  $K$ : 1, 3 e 5 a fim de encontrar o melhor resultado possível.

### 3.4.2 Máquina vetor de suporte

A escolha deste método de classificação se dá por sua natureza de classificação binária, e por ser um classificador linear, diferentemente do KNN a classificação da hipótese não cresce de acordo com a quantidade de dados, tornando-o um método de classificação menos custoso do que o KNN.

### 3.4.3 Métricas para detecção de fadiga

Com a realização da classificação dos olhos abertos ou fechados finalizada com sucesso, é necessário o uso de métricas para calcular e predizer se o motorista se encontra em estado de fadiga ou não.

A métrica escolhida para o sistemas foi: *Percentage eye openness tracking* (PER-CLOS) é definida como a percentagem de um período de tempo já predefinido (geralmente

cerca de 20% de um minuto) em que a pessoa tem 80% ou mais dos olhos fechados (DINGES; GRACE, 1998). Essa métrica é considerada uma boa opção para calcular o nível de fadiga de um indivíduo ao longo do tempo.

A fórmula para o cálculo de PERCLOS é definida na equação 3.1 onde  $N_m$  é a quantidade total de detecções e  $N_a$  a quantidade de detecções com olhos abertos.

$$PERCLOS = \frac{N_m - N_a}{N_m} * 100\% \quad (3.1)$$

## 4 Resultados e Discussão

Neste capítulo são apresentados os resultados e a avaliação de desempenho da metodologia proposta para a detecção de fadiga a partir da classificação dos olhos.

### 4.1 Resultados dos testes dos classificadores

A base de dados utilizada (CEW) foi antes avaliada com ambos os classificadores propostos, a fim de testar a capacidade de classificação de cada um e comparar a performance dos dois. Nas Tabelas 1, 2, 3 e 4 são apresentados os resultados de classificação do SVM e KNN utilizando o próprio *dataset*.

Os valores abaixo representam a média dos resultados. Ambos os classificadores foram executados pelo método de validação cruzada com 10 *folds*. O classificador KNN foi configurado com três valores diferentes de K e o método de determinação da distância foi a distância euclidiana. O SVM foi configurado com a função de decisão *one-vs-one* pelo fato de existirem apenas duas classes.

Como esperado em um ambiente controlado (a própria base de dados possui poucas variações), a classificação utilizando o HOG como extrator de características obteve uma alta taxa de instâncias corretamente classificadas em ambos os classificadores.

Tabela 1 – Resultado da classificação utilizando o SVM

Instancias corretamente classificadas	92.203%
Instancias incorretamente classificadas	7.797%
Número total de instâncias	4040

Tabela 2 – Resultado da classificação utilizando o KNN para  $k = 1$

Instancias corretamente classificadas	89.034%
Instancias incorretamente classificadas	10.965%
Número total de instâncias	4040

Tabela 3 – Resultado da classificação utilizando o KNN para  $k = 3$

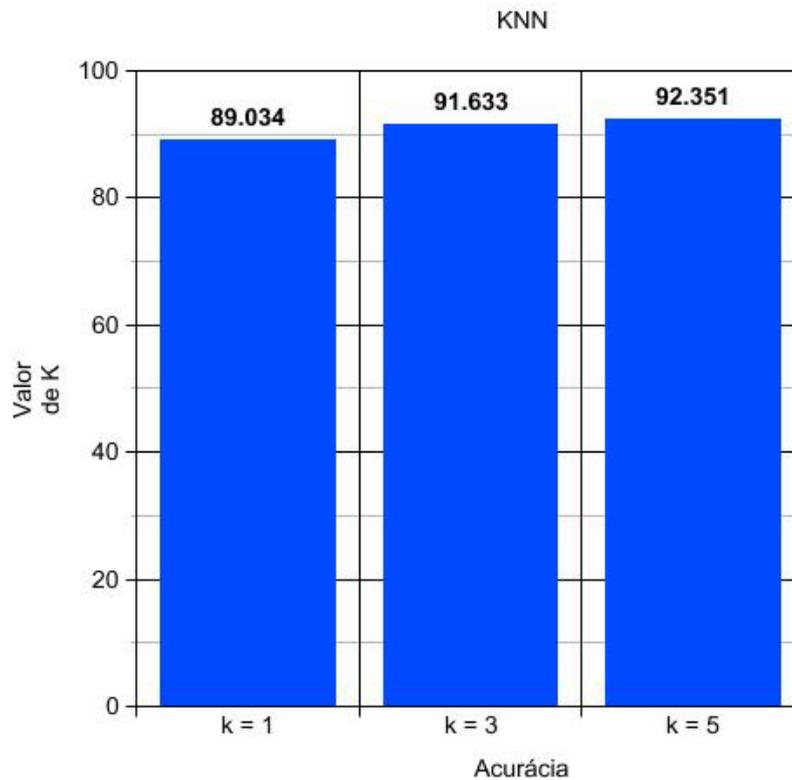
Instancias corretamente classificadas	91.633%
Instancias incorretamente classificadas	8.3663%
Número total de instâncias	4040

Também foram realizados testes com diferentes valores de K, como pode ser visualmente comparado no gráfico da Figura 13:

Tabela 4 – Resultado da classificação utilizando o KNN para  $k = 5$ 

Instancias corretamente classificadas	92.351%
Instancias incorretamente classificadas	7.6485%
Número total de instâncias	4040

Figura 13 – Comparação na acurácia para diferentes valores de K



Como pode ser observado, a acurácia do SVM foi similar ao KNN, ficando praticamente empatado para o valor de  $k = 5$ . O experimento com o KNN só foi até o valor de  $k = 5$  pois, valores maiores de K geram fronteiras menos distintas, e a taxa de erro acaba aumentando.

## 4.2 Detecção dos olhos

O processo de classificação dos estados dos olhos depende primeiramente da captura dos mesmos pelo *haar cascade* que é um classificador por si só, logo é possível que haja erros na classificação. Foi realizado um teste com 200 imagens de rostos estáticas para verificar a acurácia do classificador na detecção dos olhos. As imagens utilizadas foram obtidas da mesma base de dados. Os resultados são apresentados na Tabela 5.

Tabela 5 – Avaliação da performance do Haar cascade com imagens estáticas

Instancias corretamente classificadas	82.407%
Instancias incorretamente classificadas	17.593%
Número total de instâncias	200

### 4.3 Classificação do Nível de Fadiga

Com a imagem obtida pelo *haar cascade*, a classificação dos olhos é feita em tempo real, com cada quadro sendo enviado ao classificador que confia que a imagem é de fato, de um olho. Esses dados são contabilizados então na métrica de PERCLOS que estima com uma porcentagem o grau de fadiga do indivíduo ao longo do tempo

O resultado da classificação é mostrado diretamente na imagem, como mostra a Figura 14.

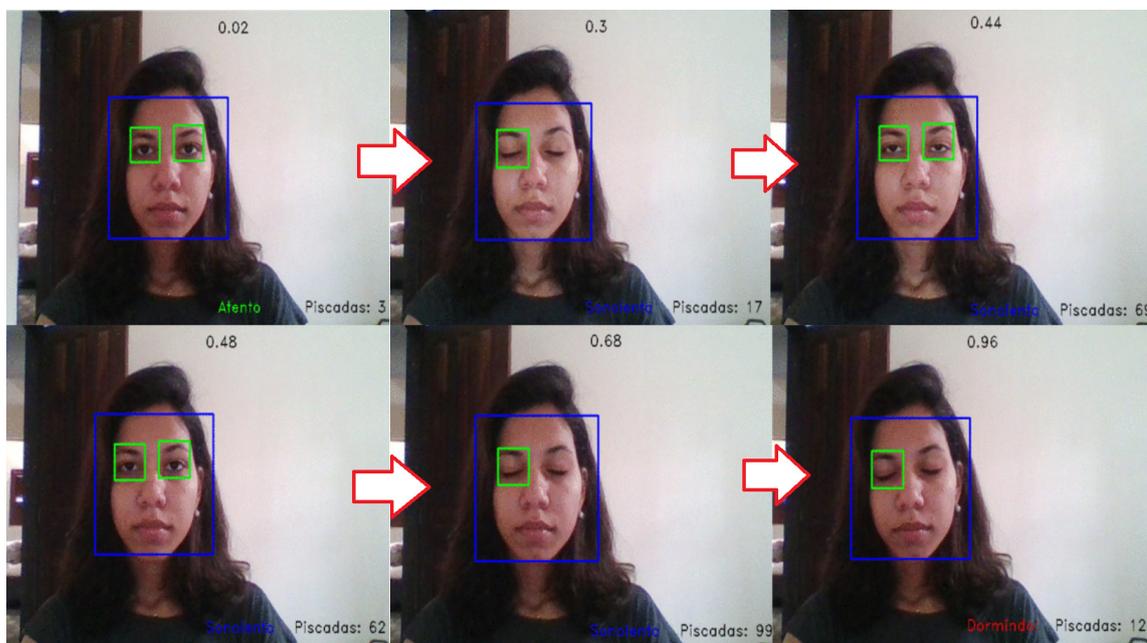


Figura 14 – Visualização da classificação em tempo real

O resultado da classificação pode ser considerado bom com ambos os classificadores, porém o SVM se mostrou ligeiramente melhor conseguindo manter uma taxa de quadros um pouco mais alta do que o KNN, mas ambos podem ser tomados como detectores em tempo real, visto que todos ficaram com uma taxa de quadros superior a 24 quadros por segundo.

A verificação de erros na classificação nesta etapa é bem difícil, dado a quantidade de quadros por segundo processados, apenas um segundo de vídeo gera 30 classificações, e em um minuto são gerados 1800 resultados de classificação. Erros provenientes da etapa de detecção dos olhos tornou uma métrica precisa extremamente difícil de se conseguir. Em geral o sistema conseguiu determinar se o indivíduo estava com os olhos abertos ou

fechados ao longo do tempo pois a maioria das detecções e classificações estavam em concordância com a porcentagem mostrada na tela pelo método de estimação de fadiga PERCLOS, porém não de forma perfeita.

## 5 Conclusão

A proposta deste trabalho era indicar se o individuo estava de olhos abertos, fechados ou piscando para inferir um estado de fadiga em tempo real. Com esses dados é possível inferir o estado do individuo em Atento, sonolento ou dormindo.

A etapa de extração de características apresentou excelentes resultados, o uso do HOG demonstrou que as características de direção dos gradientes da imagem podem ser utilizadas para diferenciar os estados os olhos, a normalização da imagem também foi de grande ajuda na etapa de classificação, diminuindo possíveis ruídos da imagem.

A extração dos olhos pelo *haar cascade* não foi perfeita, mas pode ser considerada boa. Um problema é a classificação errada de regiões, o *haar cascade* as vezes detectava regiões como o canto da boca como olhos, o que atrapalhava o processo de classificação. Por conta disso, não foi possível medir com precisão a quantidade de falsos positivos. Como os classificadores prediziam o estado dos olhos, por conta de a imagem fornecida não ser um olho a classificação era atribuída a uma das classes, causando uma classificação falsamente positiva.

Em alguns casos, ocorreram a classificação errônea dos estados do olho. No entanto, o método de inferência de estados, o PERCLOS não era gravemente afetado, visto que, em apenas 1 segundo, 30 classificações eram feitas e dentro de um período de 60 segundos, a classificação não era muito afetada por falsos positivos (não muito constantes).

Outro ponto que poderia influenciar na acurácia é o uso de outros métodos de extração de características além do HOG, pois apenas este método pode ser insuficiente para obtenção de uma classificação precisa. Algumas técnicas de processamento de imagem poderiam ser testadas, a fim de reduzir ainda mais a variância de luz ou ruídos na imagem.

Não foi possível determinar com precisão se a classificação em tempo real teve uma alta taxa de acertos pelo fato do próprio método de detecção dos olhos (*haar cascade*) ter a possibilidade de entregar falsos positivos. Talvez a escolha de um outro método de detecção dos olhos mais preciso possa limitar o problema de classificação apenas a última etapa.

## Referências

- ALTMAN, N. S. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, Taylor & Francis, v. 46, n. 3, p. 175–185, 1992. Citado na página 18.
- CBS. 2016. Disponível em: <<https://www.cbsnews.com/news/aaa-study-drowsy-driving-dangers-comparable-to-drunk-driving/>>. Citado na página 13.
- COMPASS. 2015. Disponível em: <[http://81.47.175.201/compass/index.php?option=com\\_content&view=article&id=506:413-driver-drowsiness-detection-system-for-cars&catid=22:smart-cars](http://81.47.175.201/compass/index.php?option=com_content&view=article&id=506:413-driver-drowsiness-detection-system-for-cars&catid=22:smart-cars)>. Citado na página 13.
- CORTES, C.; VAPNIK, V. Support-vector networks. *Machine learning*, Springer, v. 20, n. 3, p. 273–297, 1995. Citado na página 19.
- DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: IEEE. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. [S.l.], 2005. v. 1, p. 886–893. Citado na página 16.
- DALAL, N.; TRIGGS, B. Object detection using histograms of oriented gradients. In: *Pascal VOC Workshop, ECCV*. [S.l.: s.n.], 2006. Citado na página 16.
- DINGES, D. F.; GRACE, R. Perclos: A valid psychophysiological measure of alertness as assessed by psychomotor vigilance. *US Department of Transportation, Federal Highway Administration, Publication Number FHWA-MCRT-98-006*, 1998. Citado na página 26.
- FATIGUE, D. Road accidents: A literature review and position paper. *The Royal Society for the Prevention of Accidents*, 2001. Citado 2 vezes nas páginas 10 e 13.
- GARCÍA, I. et al. Vision-based drowsiness detector for a realistic driving simulator. In: IEEE. *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*. [S.l.], 2010. p. 887–894. Citado na página 10.
- NYT. 1997. Disponível em: <<http://www.nytimes.com/1997/09/21/us/number-of-cars-is-growing-faster-than-human-population.html>>. Citado na página 13.
- PAPAGEORGIOU, C. P.; OREN, M.; POGGIO, T. A general framework for object detection. In: IEEE. *Computer vision, 1998. sixth international conference on*. [S.l.], 1998. p. 555–562. Citado na página 13.
- ROQUE, P. M. d. S. *Técnicas de visão computacional para a deteção automática de padrões de fadiga*. Dissertação (Mestrado) — Universidade de Évora, 2013. Citado 2 vezes nas páginas 10 e 11.
- RUSSELL, S.; NORVIG, P.; INTELLIGENCE, A. A modern approach. *Artificial Intelligence*. Prentice-Hall, Egnlewood Cliffs, v. 25, p. 27, 1995. Citado na página 18.

SILVA, L. D. F. et al. Monitoramento do estado de sonolência de motoristas de automóveis através de análise de imagens de olhos. Universidade Federal do Maranhão, 2012. Citado 2 vezes nas páginas 10 e 11.

SONG, F. et al. Eyes closeness detection from still images with multi-scale histograms of principal oriented gradients. *Pattern Recognition*, Elsevier, v. 47, n. 9, p. 2825–2838, 2014. Citado 2 vezes nas páginas 24 e 25.

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. Citado 3 vezes nas páginas 12, 14 e 15.