

Análise de Desempenho de um Sistema Embarcado de Detecção de Fadiga de Condutores

Ricardo Creonte Câmara de Meira Santos
Universidade Federal de Ouro Preto
Departamento de Computação
email: ricardocamara03@gmail.com

Ricardo Augusto Rabelo Oliveira
Universidade Federal de Ouro Preto
Departamento de Computação
email: rrabelo@gmail.com

Resumo—O avanço computacional em sistemas embarcados para veículos permitiu o uso dos mesmos para aprimorar as condições de direção e do trânsito. Neste artigo propomos um sistema embarcado para monitoramento de fadiga e sonolência de condutores. Mostramos o algoritmo utilizado para a detecção de fadiga, assim como a disposição do hardware no ambiente veicular. Por fim testamos o algoritmo em 4 plataformas diferentes e mostramos uma comparação de desempenho entre elas.

I. INTRODUÇÃO

Segurança no trânsito é uma preocupação constante na sociedade moderna. E mesmo assim as taxas de acidentes continuam elevadas, e número de vítimas aumenta a cada ano.

Atualmente, fadiga e desatenção dos condutores estão entre as principais causas de acidentes de automóveis. É muito comum ver notícias sobre acidentes graves devido a sonolência ou distração dos motoristas ao volante. Visto que grande parte dos indivíduos não são capazes de julgar seu próprio estado de sonolência e distração.

Este artigo tem como objetivo apresentar um sistema que aumenta a segurança no ambiente veicular e efetuar uma comparação de desempenho entre diferentes plataformas. Este sistema descrito neste artigo foi elaborado como uma forma de monitorar a fadiga e a atenção de condutores de veículos. Com o monitoramento da situação do condutor o sistema deve identificar situações de risco e alertar o condutor destas situações. Alertando o condutor das situações de risco proveniente de distração e sonolência podemos aumentar consideravelmente a segurança no ambiente veicular.

O sistema satisfaz as seguintes necessidades:

- Assertividade: Deve ser estudada e implementada uma forma de monitorar a atenção e fadiga do condutor com uma alta taxa de assertividade. O sistema deve manter essa taxa de assertividade independente do condutor.
- Plataforma: Após obter uma técnica com alta taxa de assertividade, deve ser feito um estudo de qual plataforma terá um melhor custo/benefício, para fazer parte do hardware do sistema. Atualmente há várias plataformas de prototipagem para sistemas embarcados no mercado. Deve ser escolhido um conjunto de plataformas e este estudo deve analisar o tempo de resposta do software em cada plataforma. Assim decidindo qual é a melhor a ser usada.

Este artigo está disposto em seis seções. A atual seção tem como objetivo introduzir o trabalho ao leitor. A segunda seção, trabalhos relacionados, mostram trabalhos que abragem o mesmo assunto e mostra ao leitor qual a diferença entre a proposta deste artigo em relação aos outros trabalhos. A terceira seção, técnicas de análise facial, mostram as técnicas usadas para a detecção de faces do sistema. A quarta, o sistema proposto, faz uma visão geral do sistema. Também apresenta as plataformas as quais foram feitos os testes, assim como explica o funcionamento do algoritmo utilizado. A quinta seção, experimentos, apresenta como foram feitos os experimentos e também os resultados atingidos. Ao final temos as conclusões do trabalho.

II. TRABALHOS RELACIONADOS

Em [1] os autores apresentam uma maneira de manter o sigilo da identidade de condutores em vídeo de condução natural. E também apresentam novas descobertas sobre o efeito deste filtro no reconhecimento facial e na estimativa de onde o olhar do motorista está focado. As gravações de vídeos de motoristas podem possuir interesse caracterizado como invasão de privacidade. A avaliação de dados de condutores em ambientes naturais tem como objetivo aumentar a segurança do condutor e não influenciá-lo na maneira de dirigir. As técnicas normais de 'de-identificação' são: 'blacking out', 'blurring' em faces ou pessoas. As quais protegem a privacidade mas dificultam a análise do comportamento do motorista. Sendo assim os autores apresentam uma nova forma de manter o sigilo da identidade, a qual deve preservar a direção do olhar do condutor. Um filtro de 'de-identificação' transforma um vídeo onde o condutor pode ser reconhecido e transforma em um vídeo onde a identidade do condutor será irreconhecível. O desafio enfrentado na 'de-identificação' é que os filtros são aplicados diretamente na região de interesse, ou seja, a face. Os autores dão exemplo dos seguintes filtros:

- Um que mantém somente os olhos e o resto da imagem fica totalmente preto.
- Um outro filtro que mantém os olhos e o resto a imagem fica suavizada.
- E um última que apaga somente os olhos, colocando uma tarja preta no local dos olhos.

Para proteger a identidade e preservar o olhar eles apresentam duas abordagens:

- Preservação em primeiro plano para estimativa do olhar. Essa abordagem consiste em detectar, rastrear e isolar os olhos e o nariz, e através destes componentes da face estimar a posição da cabeça do condutor.
- Distorção do fundo para privacidade. Uma tática e colocar tudo que está em torno da região de interesse de preto.

Em [2] os autores realizaram um estudo para investigar o impacto da visualização da incerteza de sistemas de auto-condução na confiança dos motoristas em um cenário de condução automatizada. Para um grupo de condutores foi mostrado a incerteza do sistema de direção autônoma em tempo real, e a outro grupo não foi mostrado grupo o qual os autores chamaram grupo de controle. O grupo, ao qual foi mostrado a incerteza do sistema obteve uma calibração melhor na confiança.

O trabalho foi desenvolvido para fazer uma comparação entre o comportamento dos condutores que sabem das incertezas do sistema de condução e os condutores que não foram informados dessas incertezas. O artigo tem como objetivo analisar se os condutores avisados das incertezas possuem uma confiança mais calibrada no sistema, e se eles ficam mais relaxados durante a condução e também se eles ficam mais aptos a realizar outras tarefas durante a viagem.

No artigo [3] os autores propõem um framework com múltiplas câmeras para análise de movimentos da cabeça do motorista, com ênfase em robustez e operação contínua mesmo depois de movimentos bruscos. O sistema rastreia as características da face para estimar a posição da cabeça através de um modelo 3D. Eles apresentam duas soluções que exploram as limitações que temos neste contexto, e também fazem um estudo comparativo de configurações de câmeras neste contexto.

Os autores afirmam que um sistema automático de análise do movimento da cabeça deve possuir as seguintes características:

- Automático: O sistema deve se iniciar e operar sem nenhuma intervenção do usuário.
- Rápido: O sistema deve operar em tempo real.
- Ampla faixa de operação: O sistema deve ser robusto, capaz de operar em um campo de movimentos amplos da cabeça e também rápidos.
- Invariante a luz: Deve trabalhar em várias condições luminosas.
- Invariante a pessoa: O sistema deve funcionar com motoristas diferentes.
- Tolerância a oclusão: O sistema deve funcionar quando há oclusão parcial do motorista.

O artigo propõem um estimador contínuo de movimento da cabeça com câmeras distribuídas, e o estudos de diferentes configurações das câmeras. Para estimar a posição da cabeça são propostas duas soluções, baseadas em métodos geométricos para rastreamento de características faciais. E são comparados os algoritmos para as diferentes configurações de câmeras.

Os autores montam um framework com várias câmeras colocadas no cockpit do veículo, tratando cada câmera de forma independente. Um procedimento de seleção de perspectiva provê a estimativa da posição da cabeça, analisando a dinâmica temporal e a qualidade da estimativa de cada perspectiva.

Em [4] os autores contribuem mostrando como treinar classificadores em cascata baseado em características 'haar', para detecção de olhos abertos e fechados. Mostrando a importância da base de treinamento e indicando quais são os parâmetros ótimos para a aprendizagem neste caso.

A base de treinamento utilizada pelo autores possuem faces em posições frontais, rotacionadas e inclinadas para os dois lados, assim como uma grande variedade de condições de iluminação.

O classificador treinado pelos autores possui 7000 imagens negativas de objetos comuns e cenas ao ar livre. A maior taxa de falso positivos alcançada foi de 0.7% na detecção de olhos fechados para faces rotacionadas. E a menor taxa de acerto foi de 96.8%, também para as faces rotacionadas, na detecção de olhos abertos e fechados.

III. TÉCNICAS DE ANÁLISE FACIAL

Em [5] os autores demonstram uma abordagem de aprendizagem de máquina para detecção de objetos em imagens, apta a detectar objetos de forma rápida e com um alta taxa de detecção. Este artigo possui três contribuições principais:

- A primeira contribuição é uma forma de representar as imagens a qual eles chamam de "imagem integral", que permite que os recursos utilizados pelo classificador sejam computados de forma rápida.
- A segunda é o algoritmo de aprendizagem o qual é baseado em Adaboost[6], escolhendo um número pequeno de características visuais essenciais de um conjunto maior.
- E a terceira contribuição é a proposta de um método para combinar os classificadores em uma forma de cascata, na qual cada parte da cascata representa uma partes dos classificadores, onde a complexidade vai aumentando a cada nível da cascata. Isso faz com que as partes que necessitam de menos processamento sejam computadas primeiro, e se a região processada a cada nível não se mostrar promissora à detecção o processo é abortado para a região em questão. Sendo assim as regiões não promissoras à detecção são classificadas como 'negativo' de forma rápida.

Em relação a detecção de faces o algoritmo proposto em [5] mostrou uma taxa de assertividade comparável aos melhores classificadores. E se mostrou aproximadamente 15 vezes mais rápido que as abordagens anteriores.

Em [7] Rainer Lienhart e Jochen Maydt melhoraram a performance do algoritmo proposto por Paul Viola e Michael Jones[5]. Primeiramente eles desenvolveram uma forma de treinamento, incluindo mais características no quadro de aprendizagem. Este acréscimo de informações é feito através de rotações de 45 graus no treinamento do classificador gerando assim um conhecimento de domínio adicional. Este acréscimo

de informação dos classificadores diminuiu a taxa de falsos positivos em 10%.

Algoritmos para estimar a posição da cabeça podem ser classificados nas seguintes classes: baseados em características geométricas, baseados em características de textura, e também os híbridos que são baseados em textura e formas.

Como métodos testado em laboratórios não provem uma garantia de desempenho robusto em ambientes reais, então uma avaliação em ambiente natural é necessária. Outro aspecto importante é a posição da câmera, que não deve atrapalhar a vista do condutor e deve cobrir um grande número de posições da cabeça. Outro problema a ser considerado é o fato de que as condições de luminosidade variam muito no ambiente real, por isso no sistema implementado neste artigo fizemos uso de uma câmera infra-vermelho. Sendo assim o motorista é constantemente iluminado por um farol infra-vermelho e uma câmera infra-vermelho captura as imagens do mesmo, tornando o sistema invariável diante as variações luminosas.

Nestes trabalhos não foram realizadas análises de desempenho do sistema, assim como os algoritmos não foram executados em sistemas embarcados. Neste artigo foi utilizado um classificador baseado em [5] com as melhorias descritas em [7] para detecção do rosto do condutor, e outro com as mesmas características para a detecção de olhos abertos. Então, a diferença entre os classificadores são somente as bases de dados utilizadas para treinamento (faces e olhos abertos), e os parâmetros de treinamento.

IV. O SISTEMA PROPOSTO

A. Visão Geral

Toda entrada de dados será feita através de uma câmera infra-vermelho. O sistema de monitoramento de fadiga deve possuir três características principais:

- Automático: O sistema deve iniciar e operar sem que seja necessária nenhuma intervenção do usuário. A inicialização do sistema ocorrerá sempre que o carro entrar em funcionamento, finalizando quando o carro for desligado.
- Rápido: O sistema deve ser hábil a processar os dados em tempo real. Esta característica é de extrema importância para que a identificação das situações de risco ocorra a tempo de alertar o condutor.
- Invariante a pessoa: O sistema deve funcionar para condutores diferentes e obter uma alta taxa de identificação das situações de risco para todos eles.

O sistema foi dividido em três componentes básicos, sendo uma câmera, uma unidade de processamento e um computador de bordo, como apresentado na figura 1.

A câmera funciona como dispositivo de entrada das informações sobre o condutor do veículo, captando as imagens e as enviando para a unidade de processamento processá-las. A câmera deve ser posicionada frontalmente ao condutor, para que seja possível captar a face do mesmo enquanto conduz o veículo.

A unidade de processamento processa as imagens e através de um protocolo se comunica com o computador de bordo.

Assim passando as informações sobre o estado do condutor. Este processamento tem como objetivo definir para onde está voltada a face do condutor, e se o condutor está de olhos fechados ou abertos. Ou seja, para cada imagem recebida a unidade de processamento irá identificar as condições do condutor naquele momento. Para que tais objetivos sejam alcançados serão utilizadas técnicas de processamento de imagens e reconhecimento de padrões pelo software em execução na unidade de processamento.

A unidade de processamento possui como sistema operacional um sistema linux próprio para operar em sistemas embarcados, e utiliza a biblioteca OpenCV [8] para realizar o processamento de imagens. Seu microprocessador será um processador da arquitetura ARM, visando um consumo baixo de energia já que se trata de um ambiente veicular. Atualmente há várias plataformas voltadas para o desenvolvimento de sistemas desta natureza no mercado.

O computador de bordo receberá informações da unidade de processamento e também de sensores do carro. Processando os dados recebidos simultaneamente. Estes sensores irão prover informações referentes a velocidade do veículo, e a unidade de processamento informações referentes ao estado do condutor. O objetivo do computador de bordo é processar todo o conjunto de dados recebidos e identificar situações de risco alertando o condutor. Cada situação de risco será classificada com um nível pequeno, médio ou grande de risco. Para cada nível de risco será emitido um sinal sonoro específico ao condutor.

A figura 1 mostra o posicionamento do hardware dentro do carro. Neste posicionamento proposto podemos variar a localização da unidade de processamento. O posicionamento do computador de bordo e da câmera não podem ser mudados. Já que a câmera precisa obter imagem frontais do rosto do condutor. O computador de bordo por sua vez, necessita ficar próximo ao condutor para que o condutor ouça os sinais de alerta claramente e tenha fácil acesso a ele para executar outras atividades de seu interesse.

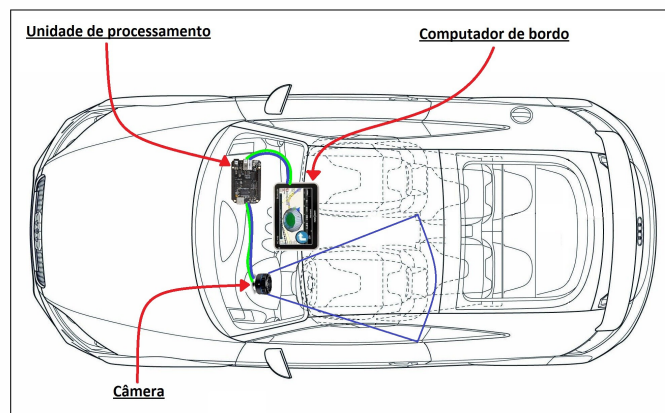


Figura 1. Posicionamento do hardware

As características faciais nem sempre estão visíveis devido à auto-occlusão, porém um sistema multi-câmera pode melhorar a robustez neste aspecto, como foi demonstrado em [4].

B. Platomas

Realizamos testes do algoritmo proposto neste artigo nas seguintes placas: BeagleBone, CubieBoard 2, WandBoard Solo e WandBoard Quad.

A BeagleBone Black possui um processador Sitara™ ARM® Cortex-A8 de 2 núcleos de 1 Ghz, 512 MB de memória RAM DDR3, acelerador gráfico 3D POWERVR SGX, 4GB de armazenamento flash interna de 8-bit eMMC, aceleração NEON de operações em ponto flutuante e uma unidade de processamento programável de tempo real (PRU-ICSS). Possui também, 1 porta USB e 1 USB OTG, entrada para cartão MicroSD, um interface Ethernet e uma saída HDMI com conector microHDMI. A alimentação pode ser através da porta USB OTG ou por uma fonte externa de 5V.

A CubieBoard 2 possui um processador ARM® Cortex™-A7 Dual-Core de 1 Ghz, um acelerador gráfico ARM® Mali400MP2, 1 GB de memória RAM DDR3, 4 GB NAND para armazenamento interno, slot para cartão micro-SD, e também, conexão SATA. A placa possui 2 hosts USB, 1 USB OTG, 96 pinos para interface incluindo I2C, SPI, RGB/LVDS, CSI/TS, FM-IN, ADC, CVBS, VGA, SPDIF-OUT, R-TP. A alimentação pode ser feita através de uma fonte de 5V ou através da porta USB OTG.

A WandBoard Solo e a WandBoard Quad possuem muitas similaridades, mas serão apresentadas em primeiro lugar as diferenças entre estas duas placas. A principal diferença são os processadores, a WandBoard Solo possui um Freescale i.MX6 Solo o qual é um Cortex-A9 Single core, enquanto a WandBoard Quad como no nome diz possui Freescale i.MX6 Quad o qual é um Cortex-A9 Quad core. A Wandboard Solo possui uma engine gráfica composta por Vivante GC 880 e Vivante GC 320, e a WandBoard Quad possui uma engine gráfica composta por Vivante GC 2000, Vivante GC 355 e Vivante GC 320. A WandBoard Solo possui 512 MB de memória RAM DDR3, enquanto a WandBoard Quad possui 2 GB. E somente a Wandboard Quad possui conexão SATA, wifi integrado e bluetooth. As semelhanças entre as duas placas são: saída de áudio, saída HDMI, 2 slots para cartões micro-SD, porta serial, header de expansão, entrada USB e USB OTG, e também interface Ethernet.

C. O Algoritmo

O algoritmo recebe como entrada uma imagem colorida adquirida pela câmera infra vermelho. Esta imagem é convertida para escala de cinza, depois é detectada a face. Caso a face esteja na posição frontal são detectados os olhos. Caso ocorra a detecção dos olhos consideramos olho aberto, caso contrário consideramos olhos fechados.

Então o software pode ser dividido em dois estágios básicos:

- Detecção da face
- Detecção dos olhos

Na detecção de face é usado o método descrito na seção anterior. Nesta etapa do processo o software utiliza dois classificadores, um treinado para detecção de faces frontais e outro treinado para a detecção de faces laterais. Caso a face frontal

seja detectada o software segue para a etapa seguinte, detecção de olhos. Caso a face frontal não for detectada o software irá procurar por faces em posições laterais. Procurando primeiro pela posição lateral a qual foi detectada pela última vez. O classificador para detecção da face voltada para esquerda e para a direita é o mesmo. Inicialmente este classificador foi treinado para detectar faces voltadas para a direita, então a imagem é espelhada para detectar faces voltadas para esquerda. Se a face lateral também não for detectada o software retorna que o condutor está distraído.

O algoritmo apresenta três estratégias para otimizar a detecção de faces. A primeira é limitar a área de busca para a área mais provável que a face do condutor se localize. Ou seja, não procurar pelo rosto em toda a imagem. Caso o rosto não se encontre nesta área de busca o software irá retornar distração do condutor. Chamamos esta área de região de busca inicial. A segunda tática para otimizar a detecção de face é limitar a área de busca da face atual por uma área próxima a face detectada anteriormente. Esta limitação será usada somente se houver detecção de face no frame anterior ao que está sendo processado no momento. Chamamos esta área de região de busca rastreada e é 188% maior que a última face detectada. A terceira tática é escalar a imagem de busca para um tamanho menor, assim fazendo com que a detecção exija menos processamento. O software escala a imagem a um tamanho equivalente a 25% do seu tamanho real.

A figura 2 demonstra o processo de detecção da face facilitando a visualização da diferença de área entre a região de busca inicial e a região de busca rastreada.

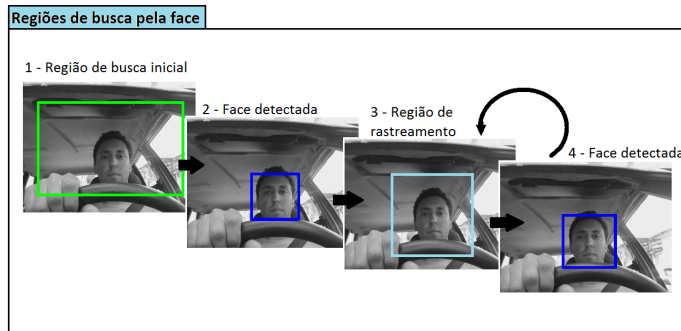


Figura 2. Regiões de busca da face

A detecção de olhos, segunda etapa do processo, recebe uma imagem contendo somente a face frontal do condutor. E busca pelos dois olhos usando um classificador treinado para detectar olhos abertos. A busca pelos olhos não ocorre em toda a imagem proveniente da etapa de detecção de faces. Com o intuito de diminuir as áreas de busca dos olhos, definimos as regiões de buscas dos olhos geometricamente, separando a área de busca do olho direito e esquerdo. Sendo assim a área de busca para cada olho é menor em relação à área total da face. O método de detecção dos olhos é o mesmo utilizado para detecção de faces, usando um classificador treinado para detectar olhos abertos e fechados.

Esta restrição das regiões de busca evita que os olhos sejam procurados nas regiões da boca, testa e bochechas. A figura 3 ilustra o processo de detecção de olhos, evidenciando as regiões de busca dos olhos.

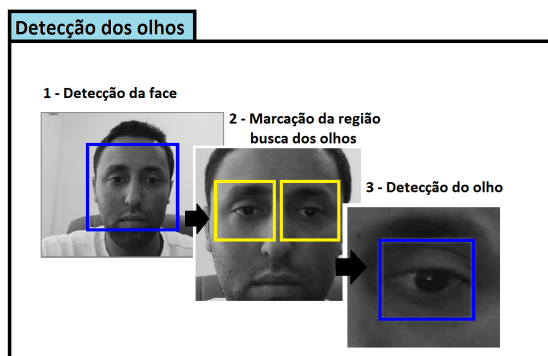


Figura 3. Regiões de busca dos olhos

Após a etapa de detecção o software tem o resultado se o olho está aberto ou fechado. Se o olho foi detectado consideramos que olho está aberto, caso contrário consideramos que o olho está fechado.

A figura 4 facilita a visualização geral do algoritmo proposto.

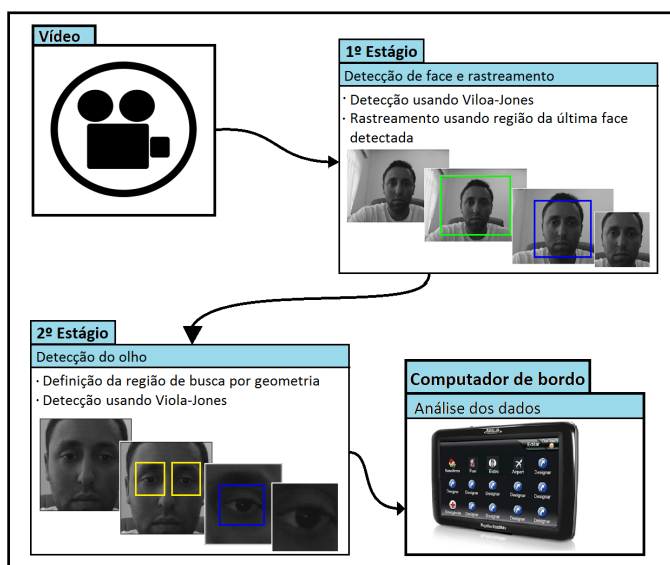


Figura 4. Algoritmo proposto

V. EXPERIMENTOS

A. Ambiente de Testes

Os experimentos foram realizados em laboratório, utilizando um cockpit de simulação de ambiente veicular. A câmera e o farol infra-vermelho foram posicionados em posição frontal em relação ao condutor, cerca de 80 centímetros de distância da face do condutor. A figura 5 mostra o ambiente de testes.

Os testes foram feitos de maneira controlada, sendo que a face do condutor esteve em posições pré-definidas durante os testes. Estas posições são frontal, esquerda e direita, e a última posição é a face voltada para baixo a qual caracteriza distração. Cada teste foi de 4 minutos de duração, e cada minuto a face do condutor esteve posicionada em uma das



Figura 5. Ambiente de Testes

posições citadas anteriormente. A ordem de posicionamento da face para cada minuto foi, frontal, esquerda, direita e para baixo. Este procedimento foi repetido para todos os hardwares descritos neste artigo, utilizando o sistema operacional Ubuntu específico para cada plataforma.

B. Resultados

A tabela 1 mostra a taxa de fps(frames por segundo) do algoritmo sendo executado em cada plataforma em relação a cada uma das posições do rosto do condutor, enquanto a tabela 2 mostra o desvio padrão.

Plataformas	Posições da Face			
	Frontal	Esquerda	Direita	Baixo
BeagleBone Black	7.10	8.95	9.11	2.13
CubieBoard 2	13.00	14.74	15.36	5.39
WandBoard Solo	15.00	13.70	13.90	4.65
WandBoard Quad	17.47	20.91	21.27	9.28

Tabela 1 - Taxas de fps

Plataformas	Posições da Face			
	Frontal	Esquerda	Direita	Baixo
BeagleBone Black	0.42	0.88	0.53	0.19
CubieBoard 2	0.70	1.31	0.85	0.15
WandBoard Solo	0.74	0.29	0.22	0.27
WandBoard Quad	0.72	1.54	1.72	0.66

Tabela 2 - Desvio Padrão

A WandBoard Quad obteve um desempenho melhor que todas as outras plataformas devido ao fato de possuir um processador com 4 núcleos. A WandBoard Solo obteve um melhor desempenho que a BeagleBone Black e a CubieBoard 2, mesmo possuindo um processador single-core.

A taxa de fps quando a face se encontra voltada para baixo foi a menor taxa, pois nessa condição o algoritmo realiza a busca da face por 3 vezes, buscando pela face frontal depois pela face voltada para esquerda e por fim pela face voltada para a direita. Assim temos a situação que exige mais processamento a cada imagem.

A taxa de fps quando a face está voltada para a direita é maior do que quando a face está voltada para a esquerda. Isso

acontece por que quando a face está voltada para esquerda o algoritmo necessita que a imagem seja espelhada antes da busca. Já que o classificador usado na busca de faces laterais foi treinado somente usando imagens de faces voltadas para a direita. Então para detectar a face voltada para esquerda é necessário mais processamento devido a necessidade de espelhar a imagem.

A taxa de fps quando a face se encontra voltada na posição frontal é menor do que as taxas para as posições laterais, exceto na WandBoard Solo. Esta taxa é menor em quase todos os casos por que quando a face está na posição frontal ocorre após a detecção a busca pelos olhos, e quando a face está em uma posição lateral esta busca por olhos não ocorre.

VI. CONCLUSÕES

Foi proposto um sistema de monitoramento de fadiga e atenção de condutores, o qual funcionará como um sistema embutido no ambiente veicular. Foi demonstrando como o software de detecção de fadiga funcionará, quais são as técnicas usadas, e como o hardware estará disposto no veículo.

O software descrito neste trabalho foi testado em diferentes plataformas de prototipagem. E foi feita uma comparação do desempenho entre estas plataformas.

Outro ponto importante é a assertividade do software proposto neste trabalho, o qual será estudado em um trabalho futuro. Testes deverão ser realizados para medir a taxa de assertividade do software. Os resultados devem ser comparados com outras técnicas de detecção, dependendo dos resultados dessa comparação outra técnica poderá ser adotada. Caso uma nova técnica for adotada serão realizados novamente testes de desempenho de hardware usando a nova técnica.

AGRADECIMENTOS

Agradecemos a Universidade Federal de Ouro Preto, ao CNPQ e a FAPEMIG. Agradecemos também à empresa SEVA Engenharia Eletrônica S/A pelo suporte ao projeto NEXUS, beneficiada pela lei da informática: 8248, 10.176 e 11.077, Brasil.

REFERÊNCIAS

- [1] S. Martin, A. Tawari e M. M. Trivedi, *Toward Privacy-Protecting Safety Systems for Naturalistic Driving Videos*. IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, 2014.
- [2] T. Helldin, G. Falkman, M. Riveiro e S. Davidsson, *Presenting system uncertainty in automotive UIs for supporting trust calibration in autonomous driving*. 5th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI ,13), Outubro 28–30, Eindhoven, Holanda, pp. 210-217,2013.
- [3] S. Martin, A. Tawari e M. M. Trivedi, *Continuous Head Movement Estimator for Driver Assistance: Issues, Algorithms, and On-Road Evaluations*. TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 15, NO. 2, APRIL, pp. 818-830, 2014.
- [4] M. Rezaei e R. Klette, *3D Cascade of Classifiers for Open and Closed Eye Detection in Driver Distraction Monitoring*. A. Berciano et al. (Eds.): CAIP 2011, LNCS 6855, pp. 171–179, 2011.
- [5] Paul Viola e Michael J. Jones, *Rapid Object Detection using a Boosted Cascade of Simple Feature*. IEEE CVPR, 2001.
- [6] Yoav Freund e Robert E. Schapire, *A decision-theoretic generalization of on-line learning and an application to boosting*. In Computational Learning Theory: Eurocolt'95, Springer-Verlag, pp 23–37, 1995.

- [7] Rainer Lienhart and Jochen Maydt, . *An Extended Set of Haar-like Features for Rapid Object Detection*. IEEE ICIP 2002, Vol. 1, pp. 900-903, Sep. 2002.
- [8] <http://opencv.org/>