Revista Brasileira de Computação Aplicada, Abril, 2021

DOI: 10.5335/rbca.v13i1.9944
Vol. 13, Nº 1, pp. 1–10
Homepage: seer.upf.br/index.php/rbca/index

ISSN 2176-6649

ARTIGO ORIGINAL

Aplicação de redes neurais convolucionais e processamento digital de imagens para classificação do estado dos olhos e avaliação de sonolência

Application of convolutional neural networks and digital image processing to classify eye state and assess drowsiness

Joany Rodrigues [1], Aline Sousa [1], Adam Santos [1]

[1]Faculdade de Computação e Engenharia Elétrica (FACEEL), Unifesspa, Unidade II - Marabá (PA) - Brasil

*joany@unifesspa.edu.br; alinefarias@unifesspa.edu.br; adamdreyton@unifesspa.edu.br

Recebido: 17/09/2019. **Revisado:** 20/04/2020. **Aceito:** 09/11/2020.

Resumo

Nos últimos anos, a quantidade de veículos que circulam nas avenidas e rodovias brasileiras tem crescido bastante. Com isso, aumentou o tempo em que as pessoas passam conduzindo seus veículos, o que ocasiona mais estresse, cansaço e falta de atenção. Em virtude dessas situações, a quantidade de acidentes também expandiu. Além disso, o ato de dirigir requer bastante atenção e disposição. Esses fatos foram relevantes para o crescimento na quantidade de acidentes, que do ano de 2016 para 2017 foi de 7.272, e aproximadamente 38% destes foram causados por condutores sonolentos. Neste trabalho, será apresentada a utilização de três técnicas de Inteligência Artificial (IA) para o desenvolvimento da aplicação em tempo real do classificador do estados dos olhos: Rede Neural Artificial (RNA) e duas Redes Neurais Convolucionais (CNN). Essas técnicas foram submetidas aos processamentos *offline* (o qual necessitou de uma base de dados com 811 fotos) e *online*. As acurácias obtidas dos processos *offline* para as três técnicas foram de aproximadamente 77% para a RNA e 95% para as CNNs. Já as acurácias dos testes *online* para a RNA, LeNet-5 e VGG16 foram 57,48%, 90,52% e 78,85%, respectivamente. Os resultados dos testes *online* mostraram que a técnica mais indicada para solucionar o problema proposto foi a LeNet-5.

Palavras-Chave: Avaliação de sonolência; Classificação do estado dos olhos; Processamento Digital de Imagem; Rede Neural Convolucional

Abstract

In recent years, the number of vehicles circulating on Brazilian avenues and highways has grown considerably. As a result, the time people spend driving their vehicles increased, which causes more stress, tiredness, and lack of attention. Due to these situations, the number of accidents has also expanded. In addition, driving requires a lot of attention and willingness. These facts were relevant to the growth in the number of accidents, which from 2016 to 2017 was 7,272, and approximately 38% of these were caused by sleepy drivers. In this work, the use of three Artificial Intelligence (AI) techniques will be highlighted for the development of the real-time application of the eye state classifier: Artificial Neural Network (RNA) and two Convolutional Neural Networks (CNN). These techniques were submitted to offline processing (which required a database with 811 photos) and online. The accuracy of the offline processes for the three techniques was approximately 77% for RNA and 95% for CNNs. The accuracy of the online tests for ANN, LeNet-5, and VGG16 were 57.48%, 90.52%, and 78.85%, respectively. The results of online tests showed that the most suitable technique for solving the proposed problem was LeNet-5.

Keywords: Convolutional Neural Networks; Digital Image Processing; Drowsiness Assessment; Eye State Classification

1 Introdução

Dormir é uma ação necessária para inibir o cansaço e a falta de atenção. O cérebro é responsável por esta ação, fornecendo estímulos que ajudam a perceber o momento adequado para iniciar o processo de repouso, fazendo com que as pessoas cumpram seu ciclo de sono. Além disso, o sono pode vir como resultado de outros fatores, por exemplo: fadiga, uso de medicamentos e estresse emocional.

A sonolência causa a redução da atenção, além de inibir o controle de músculos nos seres humanos (Neshov and Manolova, 2017). Tal fato pode influenciar negativamente em várias ações, principalmente naquelas que exigem o trabalho com máquinas, como o ato de dirigir. Apesar de não exigir muito esforço físico, dirigir é algo bastante cansativo e para um motorista fadigado e sonolento é fácil perder o controle e tornar-se incapaz de responder aos estímulos necessários para que acidentes sejam evitados.

Nos últimos anos, a quantidade de acidentes nas rodovias brasileiras cresceu consideravelmente. A Polícia Rodoviária Federal (PRF) registrou que entre 2016 e 2017, esse aumento foi de 7.272, e que aproximadamente 38% foram causados por condutores sonolentos (Polícia Rodoviária Federal, 2017). No entanto, não existe qualquer maneira legal de privar uma pessoa sonolenta de dirigir, já que não existe legislação coibitiva para motoristas sonolentos, diferentemente de legislações claras que proíbem que pessoas alcoolizadas conduzam veículos automotores.

De acordo com o exposto por Xiaoqiu et al. (2011), há conflitos entre veículos e pedestres, principalmente em avenidas altamente movimentadas. Em cenários como esses, a responsabilidade de atenção é tanto dos pedestres como dos condutores responsáveis pelo movimento dos veículos nestas avenidas. Os motoristas são responsáveis por completar ações requeridas de acordo com variáveis como: velocidade, distância e desempenho do veículo. As decisões dos motoristas para execução de manobras são os maiores causadores de impactos na segurança do trânsito, já que decisões errôneas podem ocasionar diversos acidentes fatais.

O comportamento dos motoristas é a causa de grande parte dos acidentes que ocorrem com veículos motorizados (Carvalho et al., 2017). Em 2017, de acordo com o balanço de atividades da PRF, o número de acidentes em rodovias federais foi de 89.318, que resultaram em 6.244 mortes e 83.978 feridos. A PRF também informa que a maior causa dos acidentes foi a falta de atenção, ocasionando 34.406 acidentes, com 1.844 óbitos (Polícia Rodoviária Federal, 2017). Os custos sociais de acidentes de trânsito em rodovias federais foram de aproximadamente R$ 8,9 bilhões (Polícia Rodoviária Federal, 2018). Além das vítimas, esse fato possui impactos sociais, perda de produtividade, custos médicos, legais, jurídicos e outros.

Com o objetivo de amenizar as preocupantes estatísticas anteriormente apresentadas, tem-se investido no desenvolvimento de pesquisas e aplicação de métodos que possam evitar que os motoristas, por motivo de sonolência, percam a atenção enquanto dirigem. Existem várias maneiras de detectar a sonolência, como: frequência cardíaca, identificação de desvio nas avenidas, fechamento dos olhos e da boca. Para averiguar esses parâmetros, são utilizados diversos tipos de tecnologias existentes no mercado, tais como sensores e câmeras.

Há na literatura trabalhos que propõem soluções para o problema em questão, utilizando processo de coleta de dados automático e aplicando um modelo computacional para fornecer segurança ao motorista. Essas soluções são normalmente conhecidas como sistemas de monitoramento para evitar acidentes ocasionados por motoristas sonolentos. Isso pode ser realizado por sensores e atuadores, ou ainda utilizando câmeras de monitoramento, assim como técnicas de inteligência artificial (IA) e processamento digital de imagens (PDI) (Carvalho et al., 2017, Bhoyar and Sawalkar, 2019).

Foram publicados trabalhos na literatura com o objetivo de criar sistemas ou aplicativos para detectar sonolência e utilizar essa informação de modo a alertar motorista acerca da eminência de um possível acidente, por falta de atenção (Xiao et al., 2019). As metodologias utilizam a classificação do estado dos olhos e análise do fechamento dos mesmos, fazendo a detecção e rastreamento para localizá-los e até mesmo rastrear seus movimentos. Esses procedimentos podem ser feitos em tempo real e utilizam ferramentas como: algoritmo Viola-Jones para detectar o objeto; método de descida supervisionado (*Supervised Descent Method*) para rastrear as características da face; e máquina de vetores de suporte (*Support Vector Machine*) para detectar sonolência (Neshov and Manolova, 2017, Riztiane et al., 2017, Nur et al., 2016, Hashemi et al., 2020, Geng et al., 2019). Essas técnicas foram usadas para se concluir a detecção de sonolência, no entanto existem técnicas mais atuais que podem melhorar os resultados e eficiência ao realizar aplicações para auxiliar na detecção de sonolência, como as redes reurais convolucionais (CNNs).

As análises deste trabalho se baseiam no crescente aumento do número de acidentes apresentados pela Polícia Rodoviária Federal (2017) e na importância de que ações humanas são controladas pelo cérebro, como a dormência, de forma que este obriga o corpo ao repouso, fazendo-se cumprir o ciclo de sono quando este é negado (Neshov and Manolova, 2017).

Considerando a importância do estado dos olhos do motorista, dentre outros fatores que caracterizam a sonolência, este trabalho apresenta o desenvolvimento de uma aplicação em tempo real utilizando técnicas de IA: RNA e CNNs para classificar duas fases dos olhos (abertos e fechados), usando PDI para rastrear a face. A classificação dos olhos sugere sonolência, se os olhos estiverem fechados. Imagens foram coletadas para o treinamento offline das redes neurais e o teste dessas redes foi realizado em tempo real. O desempenho das redes neurais é avaliado através da acurácia de classificação.

É importante ressaltar que este estudo apresenta apenas a avaliação do estado dos olhos, porém para uma análise mais completa do estado de sonolência, pode-se utilizar outros aspectos contribuintes, citados anteriormente.

2 Classificadores

Esta seção retrata conceitos básicos de técnicas de IA. Primeiramente, uma breve apresentação referente a RNA. Em seguida, serão conceituadas as CNNs: LeNet-5 e VGG16.

2.1 Rede neural artificial

De maneira geral, uma rede neural é uma máquina que se assemelha a um cérebro humano, criada para modelar a maneira como este realiza tarefas. Pode ser simulada por programas ou implementada a partir de componentes eletrônicos. Em sua criação, é empregado uma interligação maciça de células computacionais, conhecida por "neurônios" ou "unidade de processamento" (Haykin, 2007).

O algoritmo de aprendizagem realiza o processo de treinamento, no qual os pesos sinápticos da rede são modificados, organizando-os com o objetivo de alcançar o projeto desejado. Além dos pesos sinápticos, pode-se ainda modificar a topologia da rede.

Um dos benefícios das redes neurais é a generalização, em que são produzidas saídas adequadas para entradas não existentes na fase de treinamento (aprendizagem).

Um neurônio é a unidade de processamento de informação das RNAs, formando a base para uma rede neural. Este, contém três elementos básicos: sinápses, somador e a função de ativação (Haykin, 2007).

2.2 Redes neurais convolucionais

As CNNs são redes especializadas para processamento de dados com uma topologia que se assemelha a uma grade. Têm dados de séries temporais, que podem ser consideradas como grades $1D$ com intervalos de tempos regulares, e dados de imagens considerados como grades $2D$ de pixels (Goodfellow et al., 2016). O próprio nome da rede implica em uma operação matemática conhecida como convolução (tipo de operação linear) Haykin and Veen (2003), ou seja, são redes neurais que utilizam a convolução ao invés da multiplicação de matrizes em pelo menos uma camada.

Essas arquiteturas possuem estruturas padronizadas: camadas convolucionais padronizadas (sendo opcional a utilização de normalização de contraste e *max-pooling* em suas sequências) seguidas por uma ou mais camadas totalmente conectadas. Essas estruturas tem prevalecido em relação a problemas de classificação de imagens, produzindo resultados relativamente bons, como o desafio de classificação ImageNet (Krizhevsky et al., 2017). Para problemas como esse, com grande quantidade de dados, tem-se requerido aumentar a quantidade de camadas, assim como o tamanho dessas. Além disso, para evitar o *overfitting*, usa-se *dropout* (Szegedy et al., 2015).

As CNNs têm pelo menos duas vantagens importantes: capacidade de extrair características relevantes através de aprendizado de transformações (kernels) e depender de menor número de parâmetros de ajustes do que redes totalmente conectadas com o mesmo número de camadas ocultas. Como cada unidade de uma camada não é conectada com todas as unidades da camada seguinte, há menos pesos para serem atualizados, facilitando assim o treinamento (Szegedy et al., 2015).

Cada camada de uma CNN pode apresentar três estágios: a princípio, são executadas várias convoluções em paralelo, produzindo um conjunto de ativações lineares; em seguida, cada uma dessas ativações lineares servem de entrada para uma função de ativação não-linear. Esse processo é também chamado estágio de detector; e por fim, uma função de *pooling* é utilizada para que a saída seja modificada ou reduzida.

A função *Pooling*, substitui a saída da rede em um determinado local, por uma estatística resumida das saídas próximas (Goodfellow et al., 2016). Com isso, a utilização dessa função, auxilia na invariância de pequenas traduções da entrada, ou seja, se a entrada for traduzida por uma pequena quantidade, a maioria dos valores das saídas não serão alterados.

Deep Learning requer a capacidade de aprender automaticamente características a partir dos dados, o que geralmente só é possível quando muitos dados de treinamento estão disponíveis - especialmente para problemas em que as amostras de entrada são multidimensionais, como imagens. Dessa forma, uma solução viável é aumentar a quantidade dos exemplos utilizados, através de várias transformações aleatórias, para evitar que o modelo sobreajuste um determinado conjunto de imagens, o que pode ser nomeado por geração de dados Chollet, Francois (2016).

A seguir, serão conceituadas duas arquiteturas convolucionais.

2.2.1 *Arquitetura LeNet-5*

Esta arquitetura foi projetada para reconhecer caracteres manuscritos e impressos em máquinas na década de 90. A Tabela 1 mostra um modelo da estrutura dessa arquitetura. Essa, é composta por cinco camadas convolucionais, seguidas por camadas de *max-pooling*. A segunda e quinta camadas são seguidas por camada de *dropout*; em sequência, tem-se uma camada *flatten* e duas camadas totalmente conectadas (LeCun et al., 1989).

2.2.2 *Arquitetura VGG*

Esta arquitetura foi desenvolvida em diversos formatos, entre eles a VGG-16. Este, possui 16 camadas que, inicialmente, foi projetada para receber como entrada uma imagem de tamanho 224×224. Essa versão contém 5 blocos convolucionais seguidos de camadas de *max-pooling*. O modelo finaliza com uma camada *flatten* e um conjunto de três camadas totalmente conectadas (Simonyan and Zisserman, 2015). Na Tabela 2 é mostrada uma estrutura da arquitetura VGG-16.

Tabela 1: Estrutura de uma arquitetuta LeNet-5

Tipo de camada	Saída	Parâmetro
1ª camada convolucional (Conv2D)	(32, 32, 128)	3584
1ª camada de max-pooling (MaxPooling2)	(16, 16, 128)	0
2ª camada convolucional (Conv2D)	(14, 14, 128)	147586
2ª camada de max-pooling (MaxPooling2)	(7, 7, 128)	0
1ª camada de dropout	(7, 7, 128)	0
3ª camada convolucional (Conv2D)	(6, 6, 64)	32832
3ª camada de max-pooling (MaxPooling2)	(6, 6, 64)	0
4ª camada convolucional (Conv2D)	(5, 5, 32)	8224
4ª camada de max-pooling (MaxPooling2)	(5, 5, 32)	0
5ª camada convolucional (Conv2D)	(4, 4, 32)	4128
5ª camada de max-pooling (MaxPooling2)	(4, 4, 32)	0
2ª camada de dropout	(4, 4, 32)	0
camada flatten (Flatten)	(512)	0
1ª camada totalmente conectada (Dense)	(64)	32832
2ª camada totalmente conectada (Dense)	(10)	650

Tabela 2: Estrutura de uma arquitetuta VGG-16

Tipo de camada	Saída	Parâmetro
1ª bloco e 1ª camada convolucional	(48, 48, 64)	1792
1ª bloco e 2ª camada convolucional	(48, 48, 64)	36928
1ª bloco e camada de max-pooling	(24, 24, 64)	0
2ª bloco e 1ª camada convolucional	(24, 24, 128)	73856
2ª bloco e 2ª camada convolucional	(24, 24, 128)	147584
2ª bloco e camada de max-pooling	(12, 12, 128)	0
3ª bloco e 1ª camada convolucional	(12, 12, 256)	295168
3ª bloco e 2ª camada convolucional	(12, 12, 256)	590080
3ª bloco e 3ª camada convolucional	(12, 12, 256)	590080
3ª bloco e camada de max-pooling	(6, 6, 256)	0
4ª bloco e 1ª camada convolucional	(6, 6, 512)	1180160
4ª bloco e 2ª camada convolucional	(6, 6, 512)	2359808
4ª bloco e 3ª camada convolucional	(6, 6, 512)	2359808
4ª bloco e camada de max-pooling	(3, 3, 512)	0
5ª bloco e 1ª camada convolucional	(3, 3, 512)	2359808
5ª bloco e 2ª camada convolucional	(3, 3, 512)	2359808
5ª bloco e 3ª camada convolucional	(3, 3, 512)	2359808
5ª bloco e camada de max-pooling	(1, 1, 512)	0
camada flatten	(512)	0
1ª camada totalmente conectada	(4096)	2101248
2ª camada totalmente conectada	(4096)	16781312
3ª camada totalmente conectada	(100)	409700

3 Metodologia de Classificação

3.1 Base de dados

A base de dados foi construída com 811 imagens.

Os voluntários se posicionaram em frente a dispositivos *smartphones* para que fosse possível a captura das Imagens. Foram capturadas 2 fotos de cada voluntário, uma foto com olhos fechados e outra com olhos abertos.

Após construir a base de dados, foi feito o pré-processamento para cada imagem. As imagens possuíam cenários bastante amplos, não apenas a face das pessoas. Sendo assim, percebeu-se a necessidade de edição das fotos. Essa edição foi feita manualmente, selecionando apenas a face, que é a região de interesse para o problema proposto.

Em seguida, foi feito o redimensionamento das imagens. O processamento feito nas técnicas de IA necessita de alto poder computacional, portanto diminuir a quantidade de dados, padronizando as fotos, deixando-as com a mesma qualidade (mesma dimensão) ajuda no desempenho computacional ao realizar o treinamento das redes. Desta forma, as imagens redimensionadas (todas com a mesma dimensão) foram consideradas como entrada das técnicas de IA.

As imagens utilizadas neste trabalho estão no formato RGB (*red, green, blue*), ou seja, suas dimensões devem ser consideradas multiplicando cada imagem por 3, indicando que estas têm 3 dimensões. Então, para os testes com as redes utilizadas, as imagens foram redimensionadas, deixando-as com as dimensões de $50 \times 50 \times 3$. Na Fig. 1, podem ser observadas quatro fotos; (a) Imagem não redimensionada de olhos abertos. (b) Imagem não redimensionada de olhos fechados. (c) Imagem redimensionada de olhos fechados. (d) Imagem redimensionada de olhos abertos.

A saída das técnicas de IA foi construída identificando as fotos usando *labels*. Os olhos abertos foram definidos

Figura 1: Exemplos das fotos utilizadas na base de dados: (a) e (b) mostra a qualidade das fotos capturadas e (c) e (d) mostra a qualidade das fotos redimensionadas

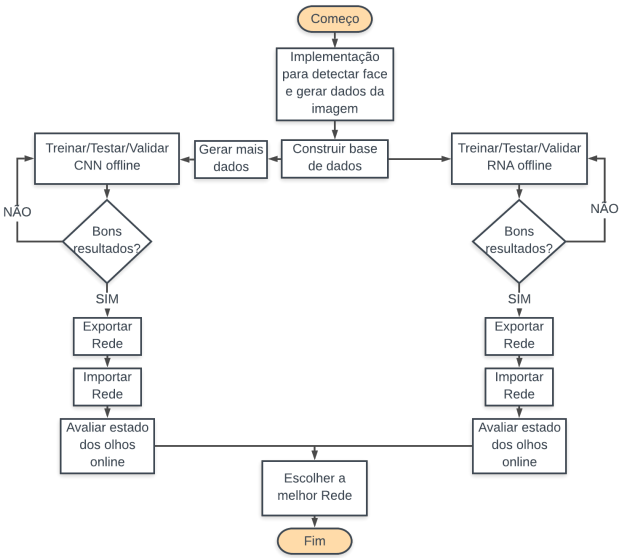

Figura 2: Atividades implementadas para o desenvolvimento do projeto

como 1 e os olhos fechados foram definidos como 0.

3.2 Etapas Realizadas para desenvolver o classificador

A Fig. 2 mostra as etapas realizadas para detecção de sonolência considerando os estados dos olhos.

A implementação em tempo real foi desenvolvida paralelamente com as outras atividades, e para o seu desenvolvimento foram utilizadas a biblioteca OpenCV e a linguagem de programação Python Howse (2013). Após a configuração dessas ferramentas, iniciou-se o desenvolvimento da aplicação, na qual são realizados processamentos utilizando PDI, como redimensionamento das imagens e detecção da face.

Juntamente com essa implementação, criou-se a base de dados, que foi necessária para o processamento offline (treinamento) das técnicas de IA. Então, com a base de dados, iniciou-se esse processamento com três arquiteturas: RNA (Pedregosa et al., 2011), LeNet-5 e VGG16 (Chollet, Francois, 2016).

Para o processamento com a RNA, a base de dados foi suficiente, mas com as CNNs necessitou-se aumentar a

Figura 3: Exemplos de fotos capturadas em tempo real: não redimensionadas (a) e (b), e redimensionadas (c) e (d)

base de dados, pois elas requerem grande quantidade de informações para realizar o treinamento offline adequadamente. Assim, a quantidade de imagens contidas na base de dados foi pouco para o treinamento das CNNs em um problema de classificação. Com isso, usou-se uma estratégia de geração (transformações aleatórias) de novas imagens a partir das imagens que já haviam na base de dados, ajudando a evitar o *overfitting* e melhorar a generalização do modelo.

A RNA utilizada foi da tipologia MLP (*multilayer perceptron*) combinada com validação cruzada Buitinck et al. (2013), objetivando escolher a melhor arquitetura de RNA. Então, a base de dados foi dividida em 80% para treinamento e validação, e 20% para teste. Em seguida, treinou-se a rede novamente com 100% da base de dados e salvou-se a arquitetura.

A estrutura das arquiteturas LeNet-5 e VGG16 foram modificadas para se adaptar à base de dados utilizada neste trabalho. Dessa forma, elas foram treinadas, testadas e validadas, de forma que a base de dados foi dividida em 60% para treinamento, e 40% para teste e validação. Então os pesos de treinamento foram exportados para serem utilizados na implementação em tempo real.

Obtendo as arquiteturas exportadas anteriormente, essas foram importadas na aplicação em tempo real. Então as imagens capturadas nessa aplicação são apresentadas às arquiteturas e estas fazem a avaliação do estado dos olhos.

3.3 Aplicação em tempo real

Para o desenvolvimento dessa aplicação, foram utilizadas a biblioteca *OpenCV* e a linguagem *Python*. A implementação foi desenvolvida para ligar a câmera do computador. Após isso, as imagens são capturadas de forma sequencial e constante, e em seguida são redimensionadas. O redimensionamento das imagens foi de $50 \times 50 \times 3$, pelo mesmo motivo que foi feito na base de dados, afinal as redes treinadas foram utilizadas nessa aplicação. Então, necessitou-se que as imagens capturadas em tempo real estivessem de acordo com o padrão das imagens que foram submetidas ao processo de treinamento das redes. A Fig. 3 mostra as imagens capturadas em tempo real, (a) imagem não redimensionada de olhos abertos, (b) imagem não redimensionada de olhos fechados, (c) imagem redimensionada de olhos abertos e (d) Imagem redimensionada de olhos fechados.

Para o redimensionamento das imagens em tempo real, foi preciso utilizar o algoritmo de detecção de face, pois o cenário das imagens capturadas é muito amplo, contendo

informações desnecessárias para o processo. Esse processo é feito com imagens em escala de cinza. Sendo assim, elas são convertidas para escalas de cinza e então a detecção é realizada, em seguida, o cenário da imagem que contêm o rosto é selecionado e este é convertida novamente para RGB, podendo assim ser redimensionada. Essa conversão para RGB ao finalizar a detecção da face foi necessária para propósitos de padronização das imagens.

Após o redimensionamento, as imagens são apresentadas às redes treinadas para fazer a avaliação do estado dos olhos, 0 ou 1, olhos fechados ou abertos, respectivamente.

4 Análise dos Resultados

Nesta seção, serão analisados os resultados das atividades mostradas na Fig. 2.

4.1 Processamento offline das arquiteturas

A utilização das técnicas de IA, teve como objetivo a avaliação do estado dos olhos, classificando-os em olhos abertos (1) ou olhos fechados (0). Então, as arquiteturas RNA, LeNet-5 e VGG16 foram submetidas ao processo offline.

Contudo, o processamento das CNNs requereu a geração de mais dados, para isso utilizou-se funções de reescalonamento, rotação, cisalhamento e zoom. Assim, foram criadas aproximadamente 2800 imagens por época a partir daquelas já existentes na base de dados, em média 2000 para treinar e 800 para validar.

4.1.1 RNA validada

Durante o processo offline da RNA, realizou-se treinamento, teste e validação. A validação é realizada no conjunto de treinamento, em que este conjunto é dividido em k conjuntos menores. Essa abordagem é conhecida como validação cruzada *k-fold*. Logo, a técnica foi treinada considerando três valores de k: 10, 7 e 5. Esses foram testados para diversas camadas escondidas, quantidades de neurônios e taxa de aprendizagem. Assim, avaliando a média do conjunto validado, o melhor resultado foi de 71% (três arquiteturas obtiveram esse valor). Então, foi escolhida a que possuía o menor desvio padrão, $0.71(+/-0.05)$, onde a acurácia para o treinamento foi de 0.862654 e para o teste foi de 0.693252. Além disso, os valores para número de neurônios na camada oculta, taxa de aprendizagem e k foram respectivamente, 1000, 0.001 e 5. Ao escolher a arquitetura, o processo foi repetido com 100% da base de dados para treinar, o qual obteve 0.765721 de acertos, essa rede foi então exportada.

4.1.2 LeNet-5 modificada

Utilizou-se como base a arquitetura LeNet-5, a qual foi modificada para realizar o processo offline. Para a sua construção, foram usadas três camadas convolucionais, cada uma seguida por camadas de ativação *ReLU* e *max-pooling*. Considerou-se a camada *flatten* sendo uma camada totalmente conectada, logo foram construídas três camadas totalmente conectadas, em que a segunda é seguida por uma camada de ativação e uma de *dropout*; o modelo é finalizado com uma camada de ativação *sigmóide*. A Tabela 3 apresenta os tipos de camadas com suas respectivas saídas

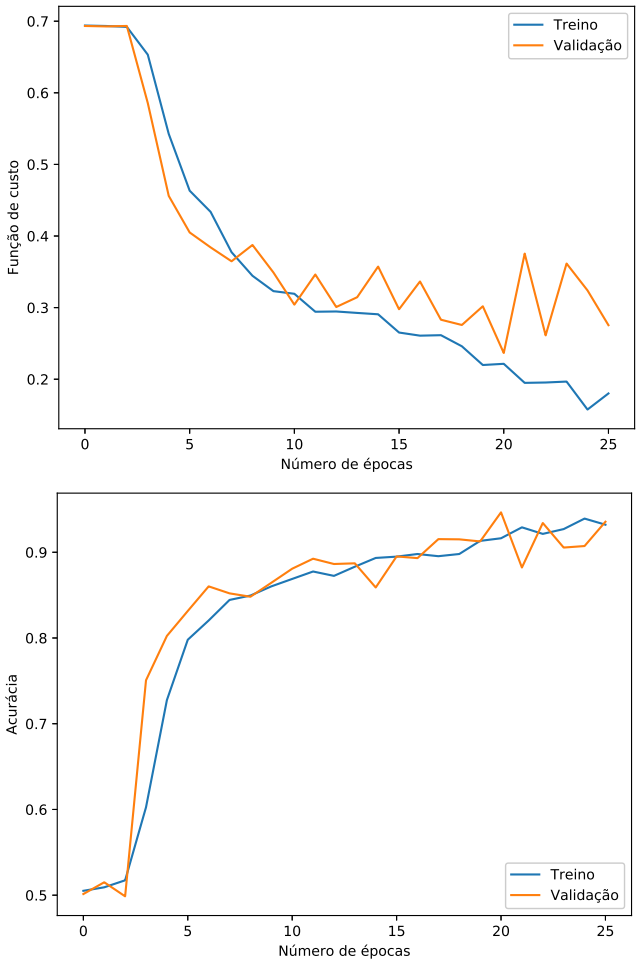

Figura 4: Processamento offline: curvas de aprendizado para o treinamento e validação (superior) e acurácia de treinamento e validação (inferior) para a arquitetura LeNet-5 modificada

e parâmetros da LeNet-5 modificada.

Com isso, o processo offline foi realizado. A função de custo e a acurácia para treinamento e validação obtidos, são apresentados na Fig. 4.

Observa-se que as curvas de função de custo estão se aproximando do valor zero, entre 0.3 e 0.2 para validação e entre 0.2 e 0.1 para treinamento. Ou seja, ao finalizar o processo, as curvas continuaram próximas uma da outra. E as curvas de acurácias de treinamento e validação seguiram aproximadas uma da outra o processo inteiro, chegando ao resultado final com aproximadamente 95%.

4.1.3 VGG16 modificada

Além da LeNet-5, a arquitetura VGG16 também foi utilizada como base e modificada para a realização do processo offline. Essa CNN foi construída com três camadas convolucionais, seguidas por camadas de *maxPooling* cada uma. Na sequência, têm-se três camadas totalmente conectadas, em que a segunda possui a ativação *ReLU* e é sequida por uma camada *dropout* e a terceira possui ativação *sig-*

Tabela 3: Camadas, saídas e parâmetros da arquitetura *LeNet-5* modificada

Tipo de camada	Saída	Parâmetro
1ª camada convolucional (Conv2D)	(48, 48, 32)	896
1ª camada de ativação (ReLU)	(48, 48, 32)	0
1ª camada de max-pooling (MaxPooling2)	(24, 24, 32)	0
2ª camada convolucional (Conv2D)	(22, 22, 32)	9248
2ª camada de ativação (ReLU)	(22, 22, 32)	0
2ª camada de max-pooling (MaxPooling2)	(11, 11, 32)	0
3ª camada convolucional (Conv2D)	(9, 9, 64)	18496
3ª camada de ativação (ReLU)	(9, 9, 64)	0
3ª camada de max-pooling (MaxPooling2)	(4, 4, 64)	0
1ª camada totalmente conectada (Flatten)	(1024)	0
2ª camada totalmente conectada (Dense)	(64)	65600
4ª camada de ativação (ReLU)	(64)	0
dropout	(64)	0
3ª camada totalmente conectada (Dense)	(1)	65
5ª camada de ativação (sigmóide)	(1)	0

Tabela 4: Camadas, saídas e parâmetros da arquitetura VGG16 modificada

Tipo de camada	Saída	Parâmetro
1ª camada convolucional (Conv2D)	(50, 50, 64)	1792
1ª camada de max-pooling (MaxPooling2)	(25, 25, 64)	0
2ª camada convolucional (Conv2D)	(25, 25, 128)	73856
2ª camada de max-pooling (MaxPooling2)	(12, 12, 128)	0
3ª camada convolucional (Conv2D)	(12, 12, 256)	295168
3ª camada de max-pooling (MaxPooling2)	(6, 6, 256)	0
1ª camada totalmente conectada (Flatten)	(9216)	0
2ª camada totalmente conectada (Dense)	(128)	1179776
dropout	(128)	0
3ª camada totalmente conectada (Dense)	(1)	129

móide. A Tabela 4 apresenta os tipos de camadas com suas respectivas saídas e parâmetros da arquitetura VGG16 modificada.

Realizou-se então o processo offline com esta técnica. A função de custo e a acurácia para treinamento e validação obtidos podem ser observados na Fig. 5.

A curva de função de custo de treinamento está abaixo do valor 0.1. Entretanto para validação, está acima de 0.3. Em mais de metade do processo as curvas se distanciaram. Já as curvas de acurácias seguiram próximas uma da outra. Ao final o resultado obtido foi aproximadamente 95% para validação e 98% para treinamento.

Os resultados offline para as duas CNNs foram os mais adequados. Porém, considerando a aproximação das curvas da função de custo e de acurácia, observa-se que a arquitetura LeNet-5 obteve melhor resultado quando comparada às demais técnicas desta análise.

Ao término do processamento offline, as arquiteturas dos dois modelos CNNs foram exportadas. Após a exportação das arquiteturas, foi feita a importação na aplicação em tempo real para que fosse possível realizar os testes online.

4.2 Processamento online das arquiteturas

Os testes foram feitos com 47 pessoas, com idades entre 18 e 75 anos, em ambientes *indoor* e *outdoor*. A distância do rosto das pessoas para a câmera foi de aproximadamente 30 cm. Foi realizada equalização de histograma, mas por não ser observada diferenças significativas com relação a diferença de luminosidade nas imagens, não foi aplicada essa equalização.

Na Tabela 5, são mostrados os resultados para cada uma das técnicas. Observa-se que entre as técnicas aplicadas, a técnica com maior percentual de acertos foi a LeNet-5.

Tabela 5: Percentual de acertos e erros para as três técnicas

Arquitetura	Acertos	Erros
RNA	57,48 %	42.52 %
LeNet-5	90.52 %	9.48 %
VGG16	78.85 %	21.15 %

Durante a realização dos testes, os voluntários foram orientados a olharem para a câmera e abrirem ou fecharem os olhos nos momentos desejados. Não houve repetição de nenhum teste. Assim, levando em consideração a posição do rosto e distância da câmera para o rosto, foi possível avaliar os testes e verificou-se que a probabilidade de acertos foi maior com a técnica LeNet-5.

Foram feitos cálculos dos testes para situações específicas. Na Tabela 6 são apresentados os resultados dos testes onde as pessoas estavam com olhos abertos e fechados. Observa-se que para olhos fechados a técnica LeNet-5 continua obtendo resultado superior, porém para olhos abertos a técnica VGG16 obteve resultado superior.

Na Tabela 7 são mostrados resultados para uma compa-

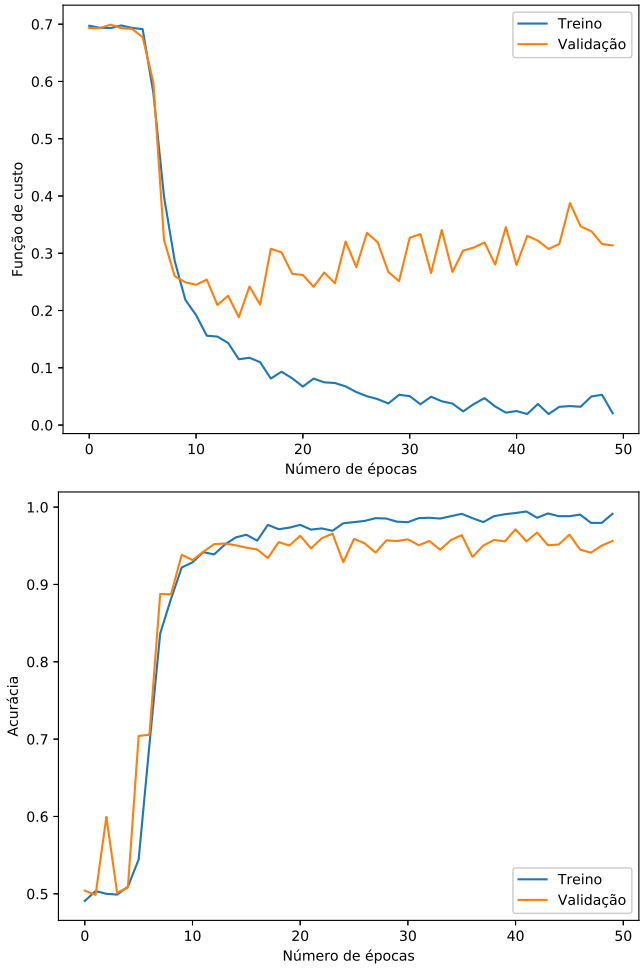

Figura 5: Processamento offline: curvas de aprendizado para o treinamento e validação (superior) e acurácia de treinamento e validação (inferior) para a arquitetura VGG16 modificada

ração onde os voluntários faziam uso ou não faziam uso de óculos. Observa-se que o percentual de acertos foi superior com a arquitetura LeNet-5.

Nas Tabelas 8 e 9 são mostrados os resultados para os testes de pessoas utilizando e não utilizando óculos, de olhos abertos e fechados. Observa-se que para olhos abertos e sem óculos o percentual de acertos obtido pela técnica VGG16 foi superior. A técnica LeNet-5 obteve melhores percentuais de acerto tanto para a análise feita com os olhos abertos e uso de óculos, quanto para a análise dos olhos fechados com e sem o uso de óculos.

Apesar da base de dados ter sido construída apenas com faces de pessoas que não utilizavam óculos, foram feitos testes com as pessoas utilizando óculos para tornar os resultados mais desafiadores. Neste caso, como pode ser observado nas Tabelas 7 a 9, o percentual de acertos dos testes foram valores superiores a 92%, valores estes que são considerados satisfatórios para essa aplicação.

É interessante salientar que o percentual de acertos foi superior para os testes em que as pessoas utilizavam óculos, quando comparado ao percentual de acertos para os testes em que as pessoas não faziam uso de óculos. Isso pode ter ocorrido devido a quantidade de testes realizados, já que apenas 8 testes foram feitos com pessoas utilizando óculos, enquanto 39 testes foram feitos com pessoas que não utilizavam óculos.

Tabela 6: Percentual de acertos dos testes em que as pessoas estavam com olhos abertos e olhos fechados.

Arquitetura	Olhos Abertos Acertos	Olhos fechados Acertos
RNA	80.84 %	34.12 %
LeNet-5	87.81 %	93.35 %
VGG16	92.37 %	64.18 %

Tabela 7: Percentual de acertos dos testes realizados com pessoas utilizando e não utilizando óculos

Arquitetura	Com óculos Acertos	Sem óculos Acertos
RNA	62.77 %	55.90 %
LeNet-5	92.59 %	90.05 %
VGG16	81.77 %	77.11 %

Foram obtidos ainda resultados de testes considerando a faixa etária dos participantes, onde conceituou-se como adultos as pessoas entre 18 e 59 anos e como idosos as pessoas com idades a partir de 60 anos. A Tabela 10 apresenta esses resultados. Observa-se que para a técnica LeNet-5 o percentual de acertos com idosos foi bastante inferior quando comparado com o percentual de acertos para os testes feitos com adultos.

Para melhor avaliação dos resultados, nas Tabelas 11 e 12 são mostrados os percentuais de acertos dos testes de adultos e idosos, com olhos abertos e fechados. Percebe-se que os resultados dos testes da técnica LeNet-5 feitos em idosos com olhos abertos foram de apenas 50%. Possivelmente esse resultado fez com que o percentual de acertos da técnica LeNet-5 tenha diminuído para os testes de olhos abertos, conforme pode ser visualizado na Tabela 6.

Observa-se que para a maioria dos resultados, incluindo os resultados gerais apresentados na Tabela 5, a arquitetura da LeNet-5 foi a que obteve um percentual de acertos maior. Um dos motivos para essa ocorrência pode ter sido os resultados do processo offline, onde a arquitetura LeNet-5 também obteve o melhor resultado quando comparada às demais técnicas utilizadas. A VGG16 apresentou as curvas de função de custo de treinamento e validação relativamente afastadas uma da outra, o que implicou em resultados inferiores ao se comparar com os resultados da LeNet-5. Se o processo continuasse ao longo da quantidade de épocas , poderia ter sido observado um *overfitting*.

Outro motivo que pode ser levado em consideração são as estruturas das redes. A LeNet-5 tem estrutura mais simplificada que a VGG16. Isto pode ter influenciado na obtenção de resultados superiores para a arquitetura LeNet-5, já que o problema não necessita necessariamente de uma arquitetura convolucional robusta, por se tratar de um problema de classificação binária.

A RNA foi a técnica com menor percentual de acertos,

Tabela 8: Percentual de acertos dos testes em que as pessoas estavam de olhos abertos utilizando e não utilizando óculos

Arquitetura	Olhos Abertos Com óculos	Olhos Abertos Sem óculos
RNA	87.97 %	78.79 %
LeNet-5	92.86 %	86.63 %
VGG16	88.57 %	93.31 %

Tabela 9: Percentual de acertos dos testes em que as pessoas estavam de olhos fechados utilizando e não utilizando óculos

Arquitetura	Olhos fechados Com óculos	Olhos fechados Sem óculos
RNA	39.01 %	32.60 %
LeNet-5	92.31 %	93.58 %
VGG16	74.49 %	61.56 %

Tabela 10: Percentual de acertos dos testes em adultos e idosos

Arquitetura	Adultos Acertos	Idosos Acertos
RNA	58.33 %	52.15 %
LeNet-5	93.51 %	68.07 %
VGG16	78.56 %	80.62 %

Tabela 11: Percentual de acertos dos testes em que as pessoas estavam com olhos abertos (adultos e idosos)

Arquitetura	Olhos abertos Adultos	Olhos abertos Idosos
RNA	79.30 %	90.36 %
LeNet-5	93.16 %	50 %
VGG16	92.75 %	89.86 %

na maioria das análises realizadas. Alguns testes, para a RNA, precisaram ser repetidos até que se conseguisse uma posição adequada, visto que a posição do rosto deveria ser bem ajustada. Isso pode ter ocorrido pelo fato de a RNA ser uma arquitetura tradicional, que não foi projetada para processar eficientemente grandes quantidades de informações (e.g, imagens), em comparação com as CNNs.

Em suma, pelos motivos supracitados, considera-se que os resultados foram satisfatórios. Das arquiteturas utilizadas para solucionar o problema proposto, a arquitetura que apresentou o maior percentual de acertos foi uma CNN de estrutura mais simplificada (LeNet-5), visto que o problema também pode ser classificado como um problema relativamente simples (problema de classificação binária).

Dos resultados obtidos neste trabalho, é possível elencar as principais contribuições deste:

- A utilização de CNNs como mais uma alternativa de ferramenta para automatizar a verificação de sonolência do condutor;
- A criação de uma base de dados com 811 fotos devidamente classificadas.

5 Conclusão

Este artigo propôs o desenvolvimento de uma aplicação em tempo real usando três técnicas de IA (RNA, LeNet-5 e VGG16) juntamente com PDI, para classificar dois estados dos olhos humanos: abertos e fechados. As técnicas foram submetidas primeiramente ao processo offline (treinamento, teste e validação) e em seguida ao processo online (testes em tempo real).

Os resultados dos processos offline e online para o objetivo de detecção de sonolência foram avaliados e, como esperado, foi demonstrado que a aplicação de CNNs possuem melhor desempenho em resoluções de problemas que requerem grande volume de informação. Sendo assim foi possível corroborar a eficácia das CNNs quando comparada a RNA para determinado tipo de problema.

A estrutura da RNA não foi projetada para processar dados multidimensionais. Acredita-se que, devido a isso, a técnica RNA não obteve resultados superiores aos resultados das CNNs.

O desenvolvimento da aplicação em tempo real é a principal contribuição deste trabalho, haja vista que esta utiliza ferramentas, como openCV e técnicas de IA para realizar o processamento dos dados de maneira não invasiva ao usuário. Outra contribuição foi a criação de uma base de dados com 811 fotos devidamente classificada em 0 (olhos fechados) e 1 (olhos abertos).

Assim, considerando o estado dos olhos como um dos parâmetros para avaliar o estado de sonolência, em trabalhos futuros pretende se analisar outros parâmetros e assim desenvolver um sistema de monitoramento de sonolência mais completo, capaz de ajudar a evitar acidentes causados por condutores sonolentos.

Referências

Bhoyar, M. A. M. and Sawalkar, S. (2019). Implementation on visual analysis of eye state using image processing for driver fatigue detection, *International Research Journal of Engineering and Technology* **6**(4): 4340–4346. https://www.irjet.net/archives/V6/i4/IRJET-V6I4950.pdf.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B. and Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project, *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122. Disponível em https://scikit-learn.org/stable/modules/cross_validation.html.

Carvalho, E., Ferreira, B. V., Ferreira, J., de Souza, C., Carvalho, H. V., Suhara, Y., Pentland, A. S. and Pessin, G. (2017). Exploiting the use of recurrent neural networks for driver behavior profiling, *in* Y. Choe and C. Jayne (eds), *2017 International Joint Conference on Neural Networks (IJCNN)*, IEEE, Anchorage, AK, USA, pp. 3016–3021. https://doi.org/10.1109/IJCNN.2017.7966230.

Chollet, Francois (2016). *Building powerful image clas-*

Tabela 12: Percentual de acertos dos testes em que as pessoas estavam com olhos fechados (adultos e idosos)

Arquitetura	Olhos fechados Adultos	Olhos fechados Idosos
RNA	37.48 %	12.5 %
LeNet-5	93.89 %	89.09 %
VGG16	63.27 %	70 %

sification models using very little data, Keras Special Interest Group. Disponível em https://blog.keras.io/building-powerful-image-classification-models-\using-very-little-data.html.

Geng, L., Hu, Z., Xiao, Z. et al. (2019). Real-time fatigue driving recognition system based on deep learning and embedded platform, *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)* **53**(1): 164–175. https://asrjetsjournal.org/index.php/American_Scientific_Journal/article/view/4735/1665.

Goodfellow, I., Bengio, Y. and Courville, A. (2016). *Deep Learning*, The Mit Press, Cambridge, Londres, UK.

Hashemi, M., Mirrashid, A. and Beheshti Shirazi, A. (2020). Driver safety development: Real-time driver drowsiness detection system based on convolutional neural network, *SN Computer Science* **1**(5): 1–10. https://doi.org/10.1007/s42979-020-00306-9.

Haykin, S. (2007). *Redes Neurais: Princípios e Práticas*, 2 edn, Bookman, Porto Alegre, Brasil.

Haykin, S. and Veen, V. (2003). *Sinais e Sistemas*, Bookman, Porto Alegre, Brasil.

Howse, J. (2013). *OpenCV Computer Vision with Python*, Packt Publishing, Birmingham, UK.

Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks, *Commun. ACM* **60**(6): 84–90. https://doi.org/10.1145/3065386.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition, *Neural Computation* **1**(4): 541–551. https://doi.org/10.1162/neco.1989.1.4.541.

Neshov, N. and Manolova, A. (2017). Drowsiness monitoring in real-time based on supervised descent method, *in* A. Sachenko and K. H. Adjallah (eds), *2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Vol. 2, IEEE, Bucharest, Romania, pp. 660–663. https://doi.org/10.1109/IDAACS.2017.8095173.

Nur, F. I. Y., Ibrahim, M. M., Manap, N. A. and Nur, S. A. (2016). Analysis of eye closure duration based on the height of iris, *in* R. Adnan and S. N. Sulaiman (eds), *2016 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, IEEE, Batu Ferringhi, Malaysia, pp. 419–424. https://doi.org/10.1109/ICCSCE.2016.7893610.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research* **12**(85): 2825–2830. http://jmlr.org/papers/v12/pedregosa11a.html.

Polícia Rodoviária Federal (2017). *PRF registra aumento de 4,8% no número de infrações de trânsito nas rodovias federais e redução de 7,5% no número de acidentes em 2017*, Assessoria Nacional de Comunicação Social - PRF, Brasília, Brasil. Disponível em https://www.prf.gov.br/portal/sala-de-imprensa/releases-1/balanco-prf-2017/view.

Polícia Rodoviária Federal (2018). *Operação Rodovida 2017/18 se encerra com redução nos acidentes e mortes em rodovias federais*, Assessoria Nacional de Comunicação Social - PRF, Brasília, Brasil. Disponível em https://www.prf.gov.br/portal/sala-de-imprensa/releases-1/balanco-rodovida-2017-2018/view.

Riztiane, A., Hareva, D. H., Stefani, D. and Lukas, S. (2017). Driver drowsiness detection using visual information on android device, *in* L. W. Santoso and R. Buyya (eds), *2017 International Conference on Soft Computing, Intelligent System and Information Technology (ICSIIT)*, IEEE, Denpasar, Indonesia, pp. 283–287. https://doi.org/10.1109/ICSIIT.2017.20.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition, *in* Y. Bengio and Y. LeCun (eds), *3rd International Conference on Learning Representations, ICLR 2015*, San Diego, CA, USA. http://arxiv.org/abs/1409.1556.

Szegedy, C., , , Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. (2015). Going deeper with convolutions, *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Boston, MA, USA, pp. 1–9. https://doi.org/10.1109/CVPR.2015.7298594.

Xiao, Z., Hu, Z., Geng, L., Zhang, F., Wu, J. and Li, Y. (2019). Fatigue driving recognition network: fatigue driving recognition via convolutional neural network and long short-term memory units, *IET Intelligent Transport Systems* **13**(9): 1410–1416. https://doi.org/10.1049/iet-its.2018.5392.

Xiaoqiu, F., Jinzhang, J. and Guoqiang, Z. (2011). Impact of driving behavior on the traffic safety of highway intersection, *in* Z. Hou (ed.), *2011 Third International Conference on Measuring Technology and Mechatronics Automation*, Vol. 2, IEEE, Shangshai, China, pp. 370–373. https://doi.org/10.1109/ICMTMA.2011.379.