

# Um Estudo sobre a Detecção de Sonolência Utilizando Deep Learning

Gustavo Iquejiri Suetugo & Leandro Nunes de Castro

Laboratório de Computação Natural e Aprendizagem de Máquina (LCoN)  
Faculdade de Computação e Informática  
Universidade Presbiteriana Mackenzie  
São Paulo – SP – Brasil

`gustavo.suetugo@mackenzista.com.br`, `lnunes@mackenzie.br`

***Abstract.** This document deals with the scientific article of the Course Completion Work developed on the topic of detecting drowsiness using Deep Learning. The objective of the study is to develop an executable model capable of detecting characteristics of sleepiness in a person through Machine Learning.*

***Resumo.** O presente documento trata do artigo científico do Trabalho de Conclusão de Curso desenvolvido sobre o tema de detecção de sonolência utilizando Deep Learning. O objetivo do estudo é desenvolver um modelo executável capaz de detectar características de sonolência em uma pessoa por meio de Aprendizagem de Máquina.*

## 1. Introdução

Entre as maiores causas de acidentes de trânsito no Brasil, o condutor dormindo é o terceiro com maior índice registrado desde 2014, sendo contabilizados mais de 22.000 casos até 2020, de acordo com os estudos da Polícia Rodoviária Federal junto a Associação Brasileira de Medicina de Tráfego (ABRAMET, 2020). Tal problema também se encontra bem ativo em outros países, como é o caso dos Estados Unidos, em que, de acordo com a U.S. National Highway Traffic Safety Administration (NHTSA), cerca de 91.000 ocorrências registradas de acidentes em rodovias foram causadas pela sonolência e quase 800 mortes no ano de 2017 (NHTSA, 2019).

Enquanto existe uma lacuna temporal até a vinda em massa dos carros autônomos e a implementação de leis que viabilizem a implantação de tal tecnologia, formas de melhorar a experiência do motorista estão em constante desenvolvimento. Entre as diversas maneiras, a utilização de câmeras para detecção de sonolência no condutor é um dos recursos especulados para implementação em carros, recurso já existente em caminhões nos Estados Unidos, apresentando resultados significativos, de acordo com a Bison Transport, com uma redução em 67% de motoristas que utilizam os veículos da empresa dirigindo cansados (Tighem, 2019).

A respeito da funcionalidade de detecção de sonolência, existem diversas maneiras de classificar o estado de uma pessoa partindo desde fatores como condições da pigmentação da pele em determinadas regiões da face, até padrões e hábitos de uma pessoa com sono, que é o caso do bocejo constante e intervalos maiores de olhos fechados (Triyanti & Iridiastadi, 2017). Existem técnicas para classificar ou detectar esse estado e

aquelas baseadas em Aprendizado de Máquina (*Machine Learning*) veem se popularizando nos últimos anos (Albawi & Mohammed, 2017).

Dentro deste contexto, este projeto de pesquisa visa responder a seguinte questão: *É possível detectar características humanas de sonolência por meio de fotos e aprendizado de máquina?*

O objetivo do projeto é desenvolver um modelo capaz de detectar sonolência por meio de fotos utilizando redes profundas (*Deep Learning*). Os objetivos específicos são:

- Estudar características e comportamentos faciais humanos de sonolência, que incluem expressões, pigmentação da pele e movimentos bruscos.
- Estudar métodos, processos, técnicas e medidas para detecção de sonolência por meio da face em humanos;
- Desenvolver um modelo executável para a detecção de sonolência em pessoas através da análise de imagens de faces humanas.

O seguinte trabalho foi organizado de forma a ser desenvolvido no segundo capítulo o referencial teórico, levantando os problemas e desafios em detectar sonolência e os métodos para tal feito. O terceiro capítulo retrata de forma teórica o desenvolver da implementação do modelo de *Deep Learning* utilizando a arquitetura e processo de treinamento do *dataset* optado. Levando ao quarto capítulo que levanta a metodologia utilizada bem como os resultados obtidos treinamento e testes. Por fim, o quinto capítulo contempla a conclusão do documento e o que pode ser evoluído.

## **2. Referencial Teórico**

### **2.1. O Problema de Detecção de Sonolência**

Reconhecer e determinar sintomas físicos ou mentais em uma pessoa sempre estão passíveis de apresentarem exceções sob os resultados esperados, como o caso de *falsos positivos* (situações normais identificadas como sonolência) ou *falsos negativos* (sonolência identificada como estado normal), ainda mais utilizando somente a informação visual. Para a detecção de sonolência (DS), tais desafios estão fortemente presentes (Triyanti & Iridiastadi, 2017).

As abordagens para se detectar a condução podem ser divididas em dois grandes grupos (Triyanti & Iridiastadi, 2017): 1) o comportamento padrão do motorista quando dirigindo de forma atenta; e 2) o comportamento natural quando o indivíduo está cansado e apresentando sonolência. Destaca-se a utilização de duas categorias para enfatizar a diferença entre os dois grupos, o comportamento dos olhos e da face, sendo que para os olhos, os fatores determinísticos são frequência de fechamento, tempo de olhos fechados, e proporção dos olhos abertos; enquanto para a face, detecção de bocejo, movimentação repentina da cabeça e expressões faciais caídas (Sun, Yu, Berilla, Liu & Wu, 2011).

Há alguns comportamentos que são característicos de motoristas que estão tentando se manter acordado e em alerta, como ouvir música, cantar, movimentar a cabeça, alongar-se, exercitar a pálpebra, ou então esfregar os olhos (Triyanti & Iridiastadi, 2017). No entanto, tais comportamentos podem atrapalhar a detecção automática de sonolência. Por exemplo, enquanto a pessoa canta o algoritmo pode interpretar como bocejos, ou assim como tampar a região de monitoramento, no caso, os olhos quando a

pessoa esfrega com as mãos, tendo um lapso de tempo da análise que não foi possível detectar.

Questões como ambientação, onde a análise é feita e comportamentos singulares da pessoa podem prejudicar na detecção de sonolência, também são comuns. Por exemplo, lugares muito escuros para detectar o rosto e, por consequência, seus comportamentos, e pessoas que apresentam tiques nos olhos ou no rosto, podem ser confundidos com sinais de alerta.

Posto isso, existe um grande desafio para a detecção automática de sonolência utilizando somente recursos visuais. Porém, o uso de aprendizado de máquina e novas maneiras de detectar tais comportamentos são oportunidades para desenvolver algoritmos e modelos ainda mais elaborados e capazes de detectar sonolência com precisão e rapidez (Phan, Nguyen, Trieu, & Phan, 2021).

## 2.2. Métodos de Detecção de Sonolência

Devido ao fato do estado de sonolência estar diretamente relacionado ao cansaço e, de tal forma, diminuir significativamente a frequência cardíaca e a intensidade respiratória, um dos métodos popularmente utilizados para detectar sonolência é o uso de eletrocardiograma para captar tais informações de maneira precisa (Sun, Yu, Berilla, Liu, & Wu, 2011): quanto menor a frequência cardíaca, mais próximo do relaxamento a pessoa está, sendo tal característica um dos indícios de sonolência. A variação do batimento cardíaco pode ser influenciada pela quantidade de estímulos gerados pelo sistema nervoso simpático e parassimpático, no qual quanto maior a presença de impulsos do sistema parassimpático, mais estímulos de relaxamento a pessoa recebe.

No trabalho apresentado em (Shinar, Akselrod, Dagan, & Baharav, 2007), 34 pacientes foram monitorados para detectar e analisar suas frequências cardíacas, sendo divididos em três grupos: pacientes normais sem anormalidades no sono; pacientes com apneia obstrutiva do sono (SAOS); e pacientes com outros distúrbios de sono. No resultado, houve diferenças significativas no ritmo cardíaco por grupo. No entanto, percebeu-se um padrão em todos: em uma média de 10 minutos antes das pessoas dormirem, todos os casos tiveram um declínio na variação do batimento, podendo então interpretar que, independentemente da condição da pessoa mediante ausência ou presença de problemas de sono, o batimento médio caía um pouco antes da pessoa entrar em estado de sono, como representado na Figura 1.

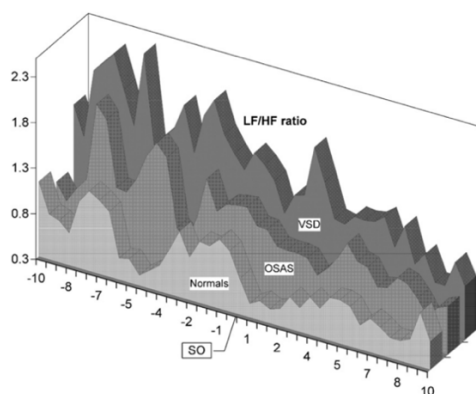
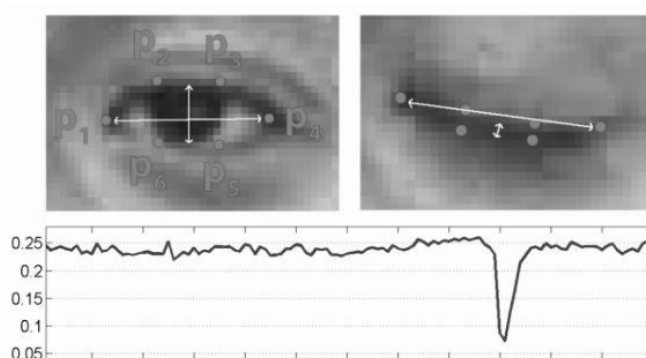


Figura 1. Gráfico de frequência cardíaca dos grupos no período de 20 minutos (Shinar, Akselrod, Dagan and Baharav 2006).

Tratando do método de observação de padrões de direção, um estudo de detecção de sonolência desenvolvido para aplicações Android fez menção aos comportamentos comuns ao dirigir um carro, como é o caso de manter-se na faixa e a postura do motorista enquanto dirige (Jabbar, et al., 2018). Enfatizando o exemplo de manter-se na faixa, de acordo com a regulamentação AEBS/LDWS-02-03 (United Nations, 2021) das Nações Unidas, para carros semiautônomos, a partir de 2029 será obrigatório o uso de sensores, tanto na traseira como na frente do carro, para detectar mudança de faixa e obstáculos, sendo que o último faz menção a objetos inanimados e seres vivos. Cabe ressaltar que tal método apresenta lacunas devido às circunstâncias, por exemplo características do veículo, condições da estrada ou habilidades e experiência do motorista, sendo tais características singulares.

No campo de métodos que fazem a análise do rosto do motorista, existem três comumente usados, sendo capazes de se complementarem para apresentar uma análise mais assertiva: monitoramento dos olhos; análise de pigmentação; e concentração de calor em região facial e análise de expressões faciais.

O monitoramento dos olhos é o método mais utilizado para detecção de sonolência (Triyanti & Iridiastadi, 2017), devido ao monitoramento estar focado na região que melhor expõe o estado de sono de uma pessoa. Tal procedimento busca encontrar os fatores padrões do comportamento dos olhos em estado acordado, para que então, caso detecte alguma anomalia no monitoramento, interprete como um comportamento de sonolência, dando ao exemplo a detecção de olhos fechados por um período mais longo que o registrado como padrão. O uso de tal método vem apresentando cada vez mais praticidade e facilidade de implementação com novos recursos capazes de detectar em tempo real sinais de sonolência, como apresentado em Soukupova & Cech (2016). Os autores utilizaram a biblioteca OpenCV no desenvolvimento do algoritmo em Python de detecção de sonolência utilizando o monitoramento do olho conforme Figura 2.

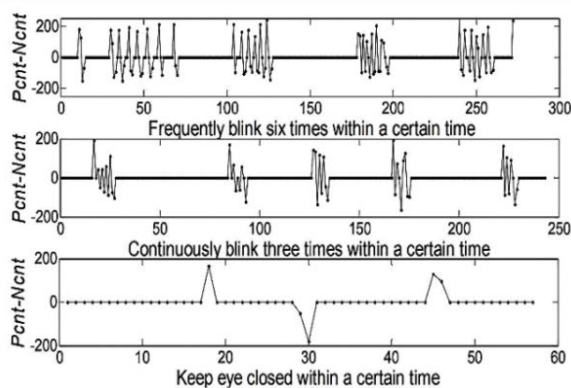


**Figura 2. Detecção de olho aberto e fechado em plano sequência (Soukupova and Cech 2016).**

Vale ressaltar que o comportamento dos olhos que dá indícios de sonolência pode variar. No entanto, pode-se determinar três grupos de comportamento predominantes: o frequente pestanejar dos olhos; o constante pestanejar dos olhos; e o tempo de olhos fechados em determinado período de tempo na Figura 3.

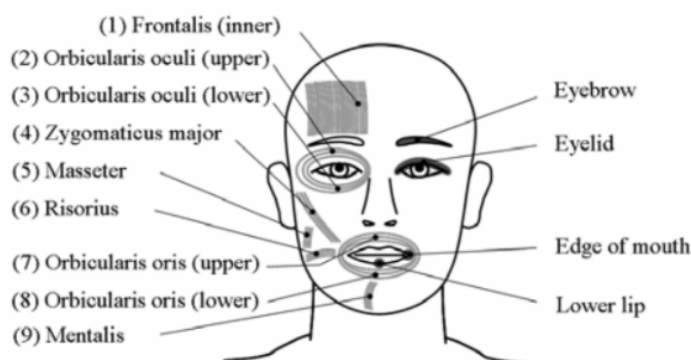
A variação de pigmentação da face também é um método utilizado para detecção automática de sonolência. Porém, há condições que acabam limitando seu desempenho, como lugares pouco iluminados e a maior precisão em pessoas caucasianas devido à detecção da pigmentação mais avermelhada na região da orelha quando em estado de

sonolência, ou um aumento de temperatura na região, indicando cansaço (Chowdhury, Shankaran, Kavakli, & Haque, 2018). Recursos mais sofisticados, como aparelhos com detecção de pontos com maior concentração de calor ou infravermelho, permitem detectar um aumento de temperatura na região da orelha, tornando o processo independente da iluminação e da cor da pele.



**Figura 3. Relação de percentual da cor gradiente ao redor do olho em um período de tempo (Agarwal and Sharma 2019).**

A face humana apresenta cerca de 11 grupos de músculos, cada um com uma finalidade diferente. Entre os grupos existem determinados músculos que apresentam uma maior variação de acordo com o estado da pessoa, como ilustrado na Figura 4 (Hachisuka, Ishida, Enya, & Kamijo, 2011). A partir desses músculos, é possível detectar não apenas o estado de cansaço, mas também o nível da sonolência através da escala de classificação Kitajima (Hachisuka, Ishida, Enya, & Kamijo, 2011). Nota-se que as expressões faciais naturais formam um dos métodos utilizados para reconhecer o estado da pessoa, como é o caso do susto, uma reação involuntária que apresenta como característica o arregalar dos olhos, ou então o estado de sonolência, que apresenta o relaxamento de músculos na região da boca, testa e olhos, apresentando uma expressão mais “caída”.



**Figura 4. Músculos com maior expressividade quando em estado de sonolência (Hachisuka, Ishida, Enya and Kamijo 2011).**

### 3. Uma Rede Convolutacional para Detecção de Sonolência

A detecção do estado de um objeto em uma imagem pode ser conduzida por meio de algoritmos de *Machine Learning* voltados para classificação, com uma variedade deles,

partindo desde modelos tradicionais utilizando somente uma camada oculta para aprendizado, assim como também os denominados Deep Learning, que além de apresentarem mais de uma camada oculta, apresentam técnicas diferentes de modelo de neurônios, e por isso podem ser mais eficazes e velozes no aprendizado (Phan, Nguyen, Trieu, & Phan, 2021).

Entre os algoritmos de Redes Profundas (*Deep Learning*), as *Redes Neurais Convolucionais* (*Convolutional Neural Network*, CNN) apresentam grande popularidade devido sua arquitetura eficaz que utiliza a lógica da operação matemática entre duas matrizes, denominada de convolução. Suas aplicações principais incluem análise de texto, áudio, imagens e vídeos, principalmente envolvendo tarefas de classificação (Albawi & Mohammed, 2017). Vale ressaltar o seu processo para detectar um objeto, que busca inicialmente todos os contornos na imagem em uma visão micro e vai se expandindo, conectando em um cenário maior todos os traços encontrados de forma interpretar e detectar o objeto na imagem da Figura 5.

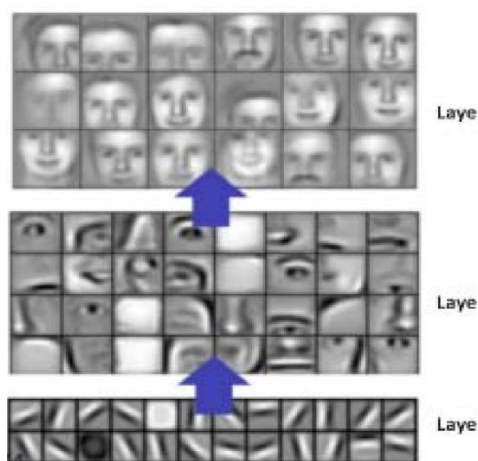


Figura 5. Camadas de aprendizado de uma CNN (Albawi & Mohammed, 2017)

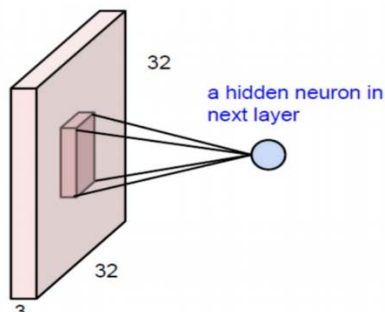
### 3.1. Modelos de Neurônios

Os diferentes tipos de neurônios em cada camada da rede exercem papéis específicos e sua escolha e ordem afetam significativamente os resultados. Assim, serão apresentados os modelos de neurônios utilizados nas várias camadas da rede CNN e suas características.

O primeiro tipo de neurônio, que recebe a entrada de dados, compõe a *camada convolucional*, que já apresenta uma grande diferença dos modelos tradicionais pela sua eficiência ao não utilizar um padrão de conexão completa (*full connection*) com a próxima camada. O foco das camadas convolucionais é em janelas de filtro, ou seja, um quadrante que vai passando pela imagem de forma a detectar e reconhecer características sem a necessidade de se preocupar com a posição do objeto na imagem. Por isso, essa camada é denominada de convolução (Albawi & Mohammed, 2017), com representação da Figura 6.

O processo da convolução em passar pelos pixels de uma imagem se chama Stride, a técnica “desliza” pela imagem utilizando a janela de filtro, com a possibilidade de trazer mais velocidade em detectar seus padrões, como contornos, que na visão computacional, são valores RGB. Isso se deve pela capacidade de mensurar o valor de pixels que a janela

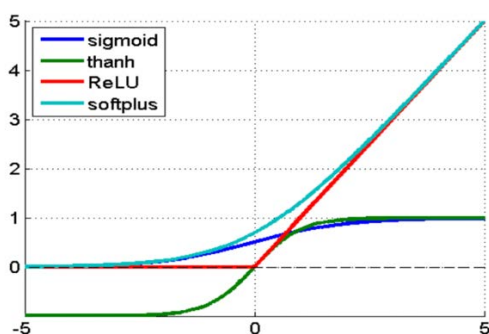
de filtro irá se mover a direita ou para baixo. Contudo ao utilizar tal técnica deve-se ter cautela, pois a quantidade de pixels que são desconsiderados a cada Stride podem ter impactos na qualidade do aprendizado.



**Figura 6. Convolução (Albawi & Mohammed, 2017)**

Da mesma forma que o processo *stride* apresenta rapidez, ele pode causar perda importante de informações ao ignorar determinados pixels. Por isso, na camada Convolutiva está presente o recurso *Zero-Padding*, que apresenta como saída o mesmo valor que a entrada, mas, para esse caso e que seja usado de fato a convolução, tal recurso pode ser usado em uma camada seguinte ao que foi utilizado no *stride*.

O próximo modelo de neurônio apresentado em redes convolucionais utiliza função de ativação não-linear. Neste caso, há vários tipos de função de ativação na literatura, como a *sigmoide*, que até pouco tempo atrás era uma das funções mais populares (Pennington, Schoenholz & Ganguli, 2017). Atualmente, a função de ativação que vem ganhando mais força, principalmente nas redes profundas, é a *Rectified Linear Unit* (ReLU), apresentando uma resposta de aprendizagem consideravelmente mais eficiente que as demais (Albawi & Mohammed, 2017), como evidenciado na Figura 7.



**Figura 7. Comparação entre os 4 modelos mais populares de função não-linear (Albawi & Mohammed, 2017).**

*Pooling* é o modelo de neurônio que visa diminuir a complexidade para as próximas camadas. Dentre os vários tipos de neurônios de *pooling*, o mais popular é o *max-poolings*, que diminui a resolução da imagem de forma a dividir em quadrantes as regiões, classificar e retornar o maior valor encontrado dentro de cada sub-região na Figura 8.

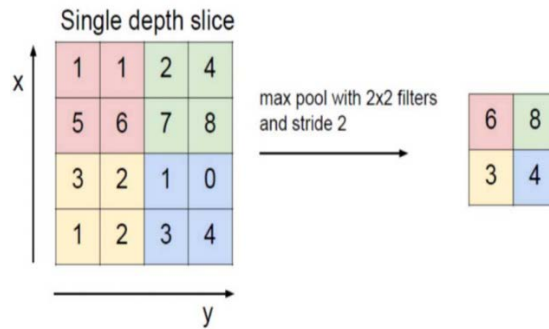


Figura 8. Ilustração de Max-pooling (Albawi & Mohammed, 2017)

Por fim, a parte final da rede é composta por uma camada *Totalmente Conectada* (*Fully-Connected*), a qual utiliza os algoritmos de treinamento típicos de redes de múltiplas camadas, como o algoritmo de *retropropagação do erro* (*backpropagation*). Esta é a camada responsável em, de fato, classificar o objeto reconhecido na imagem, sendo que a quantidade de neurônios da última camada normalmente é equivalente ao número de rótulos (classes) do problema.

### 3.2. Arquitetura do Modelo

A princípio, todas as redes CNN apresentam a mesma arquitetura básica, com as camadas convolucionais, não-lineares, *pooling* e totalmente conectadas (Albawi & Mohammed, 2017). Porém, da mesma forma que a ordem das camadas influencia o comportamento da rede, a quantidade de cada tipo de neurônio e os seus parâmetros, no caso, *stride* e tipo de função de ativação, são fatores determinantes para o desempenho da rede.

A arquitetura do modelo implementado neste projeto foi baseada, em sua grande maioria, na arquitetura *Thin MobileNet*, que apresenta resultados satisfatórios para classificação de imagens (Sinha & El-Sharkawy, 2019). Deve-se ressaltar que sua escolha para o projeto foi devida ao ambiente mais limitado onde foi executado o treinamento e testes do modelo.

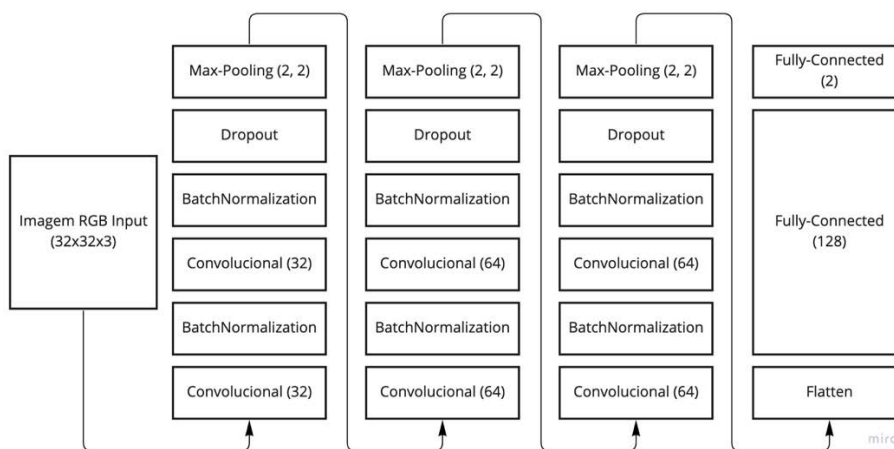
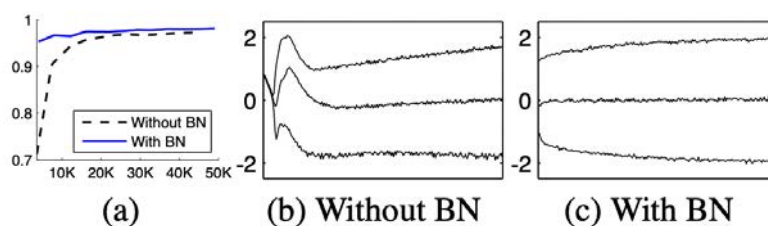


Figura 9. Arquitetura do Modelo



Dito isso, foi desenvolvida uma arquitetura utilizando as camadas convolucionais separadas, uma característica do modelo *MobileNet* (Sinha & El-Sharkawy, 2019), de forma a receber como entrada o resultado do *max-pooling* do processo anterior. A Figura 9 apresenta a arquitetura da rede implementada. É evidente a presença de etapas não mencionadas nos tipos de neurônios, como, por exemplo, a etapa de *Batch Normalization* (normalização batch, BN) e *Flatten*.

A etapa de normalização *batch* é muito utilizada em modelos de aprendizado de máquina, pois permite o uso maior de taxas de aprendizado e com um cuidado menor para a inicialização do treinamento (Ioffe & Szegedy, 2015), apresentando como principal função estabilizar as variações do processo de treinamento demonstrado na Figura 10.



**Figura 10. Teste de acurácia com e sem Batch Normalization utilizando rede MNIST (Ioffe & Szegedy, 2015).**

Os gráficos na Figura 10 representam, respectivamente, (a) a comparação do aprendizado em um modelo em treinamento com e sem uma normalização *batch*; (b) a variação de um treinamento sem uma normalização *batch*; (c) a estabilidade de um treinamento com uma normalização *batch*.

Em relação à etapa *Flatten*, que traduzida significa *achatamento*, é um processo de alinhar todos os pixels de uma imagem em uma coluna apenas, uma estrutura padrão de entrada para os modelos tradicionais, ao exemplo do MLP (MultiLayer Perceptron), assim como utilizado para entrada nas camadas totalmente conectadas.

Outra característica do modelo é a quantidade de neurônios ao longo das camadas, o qual segue uma linha lógica diferente das demais arquiteturas CNN. Neste caso, executa-se um aumento de neurônios em cada camada convolucional que é executada, partindo com 32 inicialmente e aumentando até chegar na penúltima camada totalmente-conectada com 128 neurônios e, logo em seguida, apenas 2 neurônios para classificação de *olho aberto* ou *olho fechado*.

A camada de *Max-Pooling* recebeu como parâmetros uma janela de  $2 \times 2$  e  $stride = 2$ , de forma que foi diminuída a complexidade da imagem em níveis de classificação por pixel em 4 quadrantes, em uma janela percorrendo a cada 2 pixels e tratando a saída como entrada da camada convolucional seguinte.

### 3.3. Treinamento da CNN

A respeito da implementação prática do estudo, o ambiente de treinamento e testes da rede CNN foi desenvolvido em dois notebooks processados no Google Collab, pois apresenta praticidade ao executar comandos de forma remota sem a necessidade de instalação de algum programa. Para o desenvolvimento da rede CNN, foi necessária a utilização do acelerador de hardware ligado na opção de GPU para possibilitar o treinamento de classificação das imagens. O armazenamento da base de dados, local onde

foi salvo o arquivo do modelo CNN treinado, e as imagens de teste, foram alocados no Google Drive e acessados por meio de um Ponto de Montagem no diretório raiz via linha de comando no próprio notebook.

A base de dados utilizada para treinamento foi obtida por meio do site Kaggle e é denominada “Drowsiness\_dataset” (Perumandla, 2020). Ela possui 1452 imagens de olhos abertos e fechados com dimensões variadas entre 86x86 e 84x84 pixels, sendo o primeiro notebook desenvolvido para organizar e rotular as imagens em olhos abertos ou fechados, e comprimir em um arquivo no formato .npz para maior facilidade de uso posterior.

O segundo notebook faz a normalização da base de dados em 32x32 pixels para menor complexidade e tempo de processamento, a divisão da base em treinamento (80%) e teste (20%), o processamento dos dados e a validação de resultados.

Ambos os notebooks foram desenvolvidos em Python utilizando bibliotecas, como *pandas* e *numpy* para organização e estrutura dos dados, *cv2* para análise das imagens, *matplotlib* para representação gráfica dos resultados, e *klearn*, *tensorflow* e *keras* para o desenvolvimento do modelo.

O algoritmo de treinamento se baseou no modelo MobileNet, apresentando uma arquitetura menor, com 6 camadas convolucionais, 3 max-pooling e 3 camadas fully-connected, sendo a função de ativação utilizada a *ReLU* nas primeiras camadas e *softmax* na última camada. O modelo ainda foi compilado em cima do otimizador *Adam* e recebeu um total de 226.786 parâmetros treináveis. Vale evidenciar que o algoritmo sofreu diversas modificações de forma a chegar em uma acurácia acima dos 98% para a base de dados de teste.

Para o treinamento, a base de dados ainda recebeu algumas alterações, como aumento de variações das imagens para um alcance maior de aprendizado. No caso, foram inseridas rotações de até 20° para direita ou esquerda, variações de altura e largura das imagens e rotações aleatórias de 180° em algumas imagens. Foi rodado em um ciclo de 200 épocas e com uma duração média de 45 minutos para finalizar o treinamento.

## **4. Avaliação de Desempenho**

Finalizado o treinamento, foi testado e validado o algoritmo de forma a certificar que o modelo de fato funciona, seu diferencial e limitações.

### **4.1. Metodologia**

A base de dados de teste é um conjunto pré-selecionado somente para tal função, sem qualquer pré-processamento ou alteração. As seguintes medidas de avaliação e resultados serão apresentados: os gráficos de acurácia e perda comparando aos dados de treinamento; percentual de acurácia e perda; F1-score, *recall* e precisão, e matriz de confusão.

A validação partiu da aplicação em imagens buscadas no acervo do Google, avaliando questões de pigmentação da pele, cor dos olhos, uso de óculos de grau e sol, posições do rosto, qualidade da iluminação e omissão de algumas partes da face humana. Para o reconhecimento de rosto humano foi utilizado um modelo pré-estruturado *Haarcascades* (OpenCV, 2020), que utiliza o método de detecção de objetos proposto por Viola e Jones (Viola & Jones, 2001).

## 4.2. Resultados e Discussão

Os testes apresentaram alta acurácia tanto para os dados de treinamento, quanto de teste. A Figura 11 apresenta as curvas de acurácia e perda para os dados de treinamento e teste. É possível observar que o desempenho da rede para os dados de teste foi melhor durante praticamente todas as épocas do treinamento e a rede teve um comportamento quase monotônico de melhoria de desempenho.

Da mesma forma que se obteve um percentual gradativo na acurácia, a perda se comportou de forma inversamente proporcional, ou seja, ao longo das épocas, foi diminuindo ao nível de perda um pouco menos de 0,2% conforme Figura 11, e novamente, com os dados de teste apresentando um aproveitamento maior do modelo.

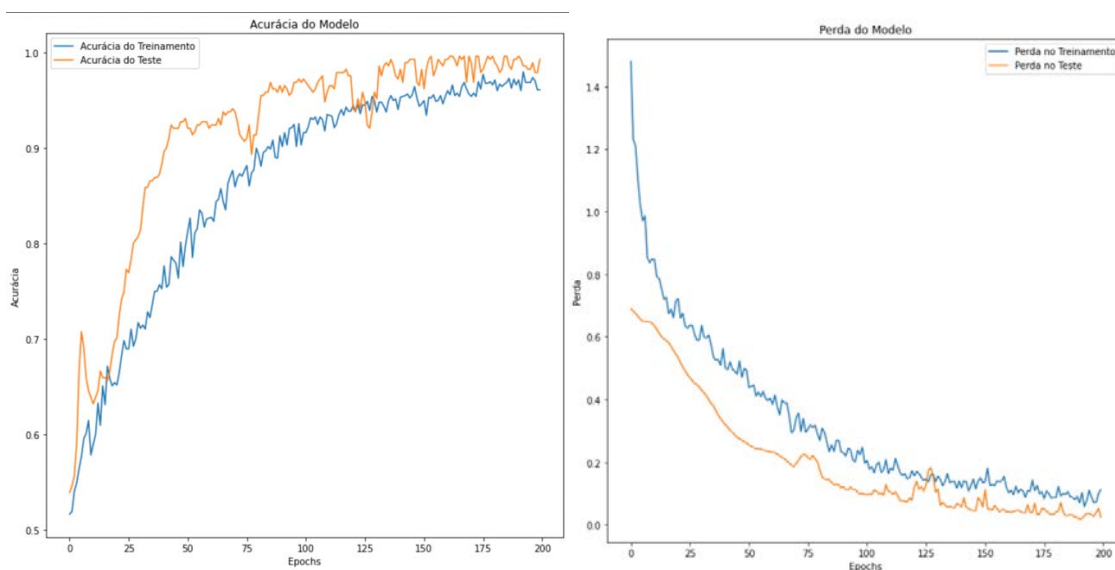


Figura 11. Gráfico de Acurácia e de Perda do Modelo

Os valores da precisão, *recall*, e F1-Score para os dados de teste estão resumidos na Tabela 1. Observa-se uma precisão média de 0,99, assim como um *recall* e F1-Score com valores 0,99. Estes valores podem ser interpretados como um modelo de alta precisão, contendo baixas taxas de *falsos positivos* e *falsos negativos*, como pode ser observado na Tabela 2 contendo a matriz de confusão.

Tabela 1. Relatório de Classificação do Modelo

	Precisão	Recall	F1-Score
Olho fechado	1,00	0,99	0,99
Olho aberto	0,99	1,00	0,99
Acurácia			0,99
Média Geral	0,99	0,99	0,99
Média dos Pesos	0,99	0,99	0,99

A matriz de confusão (Tabela 2) mostra a quantidade de objetos classificados corretamente em cada classe (olho fechado ou aberto, ou seja, verdadeiros positivos e negativos), assim como a quantidade de falsos positivos e negativos. Sendo que a

representação rotula, para os olhos fechados, o valor igual a zero e, para os olhos abertos, o valor igual a um.

**Tabela 2. Matriz de Confusão do Modelo**

	Verdadeiro Positivo	Falso Positivo
Verdadeiro Negativo	141	2
Falso Negativo	0	148

A validação prática do modelo foi feita por meio de imagens encontradas no Google, tendo sido feitos 60 testes (Tabela 3), 10 para cada categoria de teste e os seguintes resultados foram obtidos:

- i) *Pigmentação da pele*: As imagens relacionadas à pigmentação da pele foram todas reconhecidas com êxito, de forma a não impactar o desempenho do modelo;
- ii) *Cor dos olhos*: Em relação ao teste voltado para a cor dos olhos, também não houve falsos positivos ou negativos, inclusive foram testadas fotos com pessoas piscando somente com um olho, as quais também foram classificadas corretamente;
- iii) *Uso de óculos de grau e sol*: Para fotos com a presença de óculos, foram testadas 5 para óculos escuros e 5 para óculos de grau. Nestes testes houve as maiores imprecisões, pois por parte dos óculos de grau, alguns apresentavam reflexo na lente, o que provocava um fator anormal para o modelo, caracterizando sempre como um olho aberto. Para as imagens de óculos de sol, como não era possível reconhecer o olho das pessoas, a rede também não era capaz de identificar se o olho estava aberto ou fechado, gerando falsos positivos da mesma forma;
- iv) *Posições do rosto*: As fotos que apresentavam posições do rosto invertidas ou então inclinadas foram classificadas corretamente devido ao aumento de dados para o treinamento, sendo capaz de reconhecer em ângulos diferentes um olho humano;
- v) *Qualidade da iluminação*: No entanto, para locais com pouca iluminação, o modelo apresentou algumas falhas inclusive em fotos que apresentavam iluminação irregular, ou seja, em alguns pontos mais claros e em outros mais escuro, causando um sombreamento irregular em regiões faciais que impossibilitavam o reconhecimento do rosto e, portanto, sem a possibilidade de classificar se a pessoa estava com o olho aberto ou fechado;
- vi) *Omissão de algumas partes da face humana*: Como última validação utilizando imagens, foram utilizadas algumas fotos com véus, cortes de algumas partes da face e uso de máscara. Para tal cenário, foram obtidos resultados falhos novamente, em que para as fotos com véu, mostrando somente o olho apresentou dificuldades em captar o rosto e por isso apresentando algumas vezes falsos negativos, se enquadrando nessa mesma questão fotos com máscaras e imagens com cortes bruscos omitindo algumas partes do rosto.

**Tabela 3. Percentual e valor absoluto dos testes de imagens no Modelo**

	Sucedido	Falhado
Pigmentação da pele	100% (10)	0% (0)
Cor dos olhos	100% (10)	0% (0)
Uso de óculos de grau e sol	30% (3)	70% (7)
Posições do rosto	100% (10)	0% (0)
Qualidade da iluminação	60% (6)	40% (4)
Omissão de algumas partes da face humana	40% (4)	60% (6)

## **5. Conclusões e Perspectivas Futuras**

O presente artigo partiu da problemática de diversos casos de acidentes de trânsito ocorrerem devido o cansaço, que por consequência leva a sonolência (Triyanti & Iridiastadi, 2017), e que levou ao objetivo do estudo, elaborar um documento voltado para o estudo da classificação de uma das características de sonolência por meio de um recurso atual e cabível de implementação. Logo após, iniciou o estudo a respeito das limitações na atual conjuntura para detectar os fatores de sonolência em uma pessoa, com a finalidade de entender os maiores desafios em compor e desenvolver uma implementação prática.

Como conclusão, o objetivo do estudo foi alcançado ao desenvolver um estudo teórico e prático, e como resultado, um modelo CNN capaz de classificar por meio de imagem de um rosto se a pessoa estava com olhos abertos ou fechados. Apresentando como pontos fortes a detecção e classificação em 100% dos casos em que o rosto aparecia por completo e com uma iluminação que não prejudicasse o desfalque dos contornos do rosto e olhos.

Conforme testes, é nítido um enfoque no desenvolvimento dos pontos fracos futuramente, como a detecção mesmos com presença de sombras e omissões de algumas partes da face humana, assim como a dificuldade de captar em situações em que há reflexos e/ou a presença de óculos com uma lente mais escura, dificultando a detecção dos olhos.

Outro ponto que vale ponderar é a questão do uso utilizando imagens ao invés de vídeo ou então uma câmera, e nesse caso para tal capacidade será necessário um aprimoramento da parte de hardware, como o caso de uma melhor placa de vídeo ou processamento gráfico, para captar em tempo real os *frames*, que no caso, são imagens em uma sequência estipulada por quadros por segundo.

Para perspectivas futuras, nota-se um grande caminho até uma aplicação de fato que interprete o comportamento de sonolência em pessoas em um aspecto mais refinado utilizando vídeo e desenvolvendo as limitações, como o caso de baixa iluminação.

## **Referências**

ABRAMET (2020) “Problemas na saúde de motoristas causaram mais de 280 mil acidentes nas rodovias desde 2014, aponta Abramet”,

- <https://www.abramet.com.br/noticias/problemas-na-saude-de-motoristas-causaram-mais-de-280-mil-acidentes-nas-rodovias-desde-2014-aponta-abramet/>, Dezembro.
- NHTSA (2019) “Drowsy driving”, <https://www.nhtsa.gov/risky-driving/drowsy-driving>, Dezembro.
- Tighem, G. (2020) “In-Cab Monitoring Uses Machine Vision to Detect Fatigued, Distracted Driving”, <https://www.ttnews.com/articles/cab-monitoring-uses-machine-vision-detect-fatigued-distracted-driving>, Dezembro.
- Triyanti, V. and Iridiastadi, H. (2017) “Challenges in detecting drowsiness based on driver’s behavior”, IOP Conference Series: Materials Science and Engineering.
- Albawi, S. and Mohammed, T. (2017) “Understanding of a Convolutional Neural Network”, The International Conference on Engineering and Technology 2017.
- Phan, A. et. al. (2021) “An Efficient Approach for Detecting Driver Drowsiness Based on Deep Learning”, Applied Sciences MDPI.
- Sun, Y. et. al. (2011) “An In-Vehicle Physiological Signal Monitoring System For Driver Fatigue Detection”, Submitted to the 3rd International Conference on Road Safety and Simulation, Indianapolis, USA.
- Shinar, Z., Baharav, A., Dagan, Y. (2006) “Autonomic changes during wake-sleep transition: A heart rate variability based approach”, *Autonomic Neuroscience: Basic & Clinical*.
- Jabbar, R. et. al. (2018) “Real-time Driver Drowsiness Detection for Android Application Using Deep Neural Networks Techniques”, The 9th International Conference on Ambient Systems, Networks, and Technologies (ANT 2018).
- United Nations (2021) “Uniform Provisions Concerning The Approval Of Motor Vehicles With Regard To The Lane Departure Warning System”, Concerning The Adoption Of Uniform Technical Prescriptions For Wheeled Vehicles, Equipment And Parts Which Can Be Fitted And/Or Be Used On Wheeled Vehicles And The Conditions For Reciprocal Recognition Of Approvals Granted On The Basis Of These Prescriptions.
- Soukupova, T. and Cech, J. (2016) “Real-Time Eye Blink Detection using Facial Landmarks”, Center for Machine Perception, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague.
- Chowdhury, A. et. al. (2018) “Sensor Applications and Physiological Features in Drivers’ Drowsiness Detection: A Review”, *IEEE Sensors*, DOI 10.1109/JSEN.2018.2807245.
- Hachisuka, S. et. al. (2011) “Facial Expression Measurement for Detecting Driver Drowsiness”, DENSO CORPORATION, Research Laboratories, Interdisciplinary Graduate School of Science and Technology, Shinshu University, Japan.
- Agarwal, P. and Sharma, R. (2019) “Driver Drowsiness Detection Techniques: Review”, *Driver Drowsiness Detection System: D3*, DOI 10.13140/RG.2.2.20014.74567.
- Pennington, J. et. Al. (2017) “Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice”, 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

- Sinha, D. and El-Sharkawy, M. (2019) “Thin MobileNet: An Enhanced MobileNet Architecture”, 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA.
- Ioffe, S. and Szegedy, C. (2015) “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”, Google, 1600 Amphitheatre Pkwy, Mountain View, CA, USA.
- Perumandla, D. (2020) “Drowsiness\_dataset”, <https://www.kaggle.com/datasets/dheerajperumandla/drowsiness-dataset> , Kaggle, Janeiro.
- OpenCV (2020) “Haarcascades”, <https://github.com/opencv/opencv/tree/master/data/haarcascades>, Open Source Computer Vision, Janeiro.
- Viola, P. and Jones, M. (2001) “Rapid Object Detection using a Boosted Cascade of Simple Features”, Accepted Conference on Computer Vision And Pattern Recognition 2001.