DETECÇÃO DE SONOLÊNCIA EM MOTORISTAS: COMPARATIVO ENTRE *MOBILENET* E CNN SIMPLES COM ANÁLISE *GRAD-CAM*

Maria A. F. Teixeira[1], Ângela M. A. Souza[1], Rosana C. B. Rego[2]

[1] *Bacharelado em Engenharia de Software, Universidade Federal Rural do Semi-Árido, Rio Grande do Norte, Brasil (maria.teixeira@alunos.ufersa.edu.brl)*

[2] *Departamento de Engenharias e Tecnologias, Universidade Federal Rural do Semi-Árido, Rio Grande do Norte, Brasil*

Resumo: A sonolência ao volante é responsável por cerca de um terço dos acidentes de transito. O presente trabalho tem como objetivo fazer uma comparação entre o desempenho da CNN simples e a *MobileNet* na detecção de sonolência em motoristas. Ambos os modelos foram treinados com um *Dataset* de 2.900 de olhos, bocas e rostos. Os resultados obtidos indicam que o *MobileNet* é o modelo mais adequado para o problema.

Palavras-chave: CNN; *MobileNet*; Sonolência; Visão Computacional; *Deep Learning*

INTRODUÇÃO

Os automóveis são um meio de transporte essencial para os dias atuais. Cada vez mais cresce a demanda por automóveis; com isso, surge a preocupação com a segurança dos motoristas. Dentre os causadores de acidentes de trânsito, encontra-se a sonolência, responsável por aproximadamente um terço dos acidentes automobilísticos (CANANI; BARRETO, 2001).

Ao longo da jornada de trabalho, é comum que o trabalhador experimente uma sensação de cansaço ao executar determinadas atividades, refletindo também na queda de produtividade. Contudo, muitos desses estados de desgaste físico e mental nem sempre são percebidos conscientemente pelo indivíduo ou, ainda que sejam, acabam sendo ignorados. Isso pode levar a ocorrência de erros humanos, provocando acidentes no ambiente de trabalho, doenças ocupacionais ou até mesmo uma redução na qualidade do que é produzido. A influência do cansaço sobre a ocorrência de acidentes pode variar amplamente, conforme a seriedade e as circunstâncias de cada incidente (COSTA et al., 2024).

Nesse contexto, a visão computacional embarcada surge como uma alternativa promissora para auxiliar na prevenção de acidentes causados por sonolência no trânsito. Essas tecnologias possibilitam a análise em tempo real do comportamento do condutor, permitindo identificar sinais de sonolência de forma automática e eficiente.

Este trabalho tem como contribuição comparar dois modelos de Redes Neurais Convolucionais (CNNs), uma CNN simples e a *MobileNet*, para a detecção de sonolência em motoristas. A escolha da CNN simples se dá por sua facilidade de implementação e baixo custo computacional. Já a *MobileNet* foi selecionada por ser uma arquitetura leve e otimizada para sistemas embarcados.

Este trabalho está organizado da seguinte forma: inicialmente, a Seção apresenta os Trabalhos Relacionados, contextualizando as pesquisas recentes sobre o uso de redes neurais convolucionais e arquiteturas leves na detecção de sonolência. A Seção detalha os Materiais e Métodos, descrevendo as etapas de coleta, pré-processamento e balanceamento da base de dados, além das configurações e arquiteturas propostas para a CNN simples e para a *MobileNet*. Na Seção , são apresentados e debatidos os Resultados e Discussão, incluindo a análise das curvas de perda, matrizes de confusão, métricas de desempenho e a aplicação da Inteligência Artificial Explicável (XAI) via Grad-CAM. Por fim, a Seção traz a Conclusão, sintetizando as contribuições do estudo, suas limitações atuais e as propostas mapeadas para trabalhos futuros.

TRABALHOS RELACIONADOS

A detecção de sonolência em tempo real é uma exigência crítica para os Sistemas Avançados de Assistência ao Motorista (ADAS). Para operar no interior de veículos, as soluções precisam equilibrar alta precisão de reconhecimento com eficiência computacional, permitindo a execução em hardware com recursos limitados na borda (*edge computing*) (GIAP et al., 2026). Nesse contexto, a literatura recente tem se voltado maciçamente para as CNNs leves e eficientes.

Entre as arquiteturas otimizadas para visão computacional embarcada, a família *MobileNet* tem sido amplamente adotada. Seu sucesso deve-se ao uso de convoluções separáveis em profundidade, que reduzem significativamente o custo computacional e o número de parâmetros sem perdas drásticas de acurácia(SANTHIYA et al., 2025). Estudos recentes validam essa eficiência: Chitteti et al. (2025) propuseram um sistema em tempo real utilizando a *MobileNet* que alcançou 94% de acurácia com uma latência de inferência de apenas 0,00829 segundo. Da mesma forma, Rajput et al. (2025) demonstraram que a aplicação de *Transfer Learning* na MobileNetV2, ajustando pesos pré-treinados, acelera o treinamento e garante alta ro-

bustez contra variações de iluminação e posicionamento, alcançando até 98,5% de acurácia em bases de dados de estados oculares.

Vimala et al. (2026), ao compararem diversas arquiteturas, reforçam que modelos da família *MobileNet* oferecem o melhor compromisso entre o desempenho de reconhecimento e a eficiência computacional, sendo altamente recomendados para sistemas de monitoramento de motoristas.

Apesar da hegemonia de modelos pré-estabelecidos, pesquisas recentes questionam se arquiteturas mais complexas são sempre necessárias. Santhiya et al. (2025) observaram que, embora modelos mais densos atinjam altas acurácias, eles exigem recursos computacionais que inviabilizam seu uso em sistemas embarcados de baixo custo, tornando redes menores muito mais atrativas. Corroborando essa visão, Otto et al. (2025) desenvolveram uma CNN ultra-leve para detecção do estado dos olhos em dispositivos com restrição de hardware, demonstrando que o aumento da complexidade e do tamanho da rede não leva, necessariamente, a um melhor desempenho em ambientes restritos. Essas conclusões motivam a exploração de CNNs rasas e projetadas para tarefas binárias específicas.

Embora a literatura valide tanto o uso de redes baseadas na *MobileNet* quanto a exploração de CNNs ultra-leves, são escassas as pesquisas que realizam comparações empíricas diretas e balanceadas focadas no impacto do tamanho do modelo frente a métricas críticas de segurança viária. O presente trabalho preenche ativamente essa lacuna ao realizar um ensaio empírico pareado. Desenvolveu-se uma CNN customizada, com apenas 265.409 parâmetros, comparada diretamente a uma arquitetura *MobileNet* adaptada via *Transfer Learning*, composta por 3.525.313 parâmetros. Ambos os modelos foram submetidos às mesmas condições de contorno e balanceamento na base yawn_eye_dataset_new.

MATERIAL E MÉTODOS

Nesta seção, serão descritos os procedimentos metodológicos adotados para detectar a sonolência de motoristas no trânsito, por meio de modelos treinados com redes neurais convolucionais. Dessa forma, apresenta-se a descrição detalhada das etapas de coleta, tratamento e análise dos dados, garantindo a consistência, reprodutibilidade e validade dos resultados.

Base de dados

Utilizou-se o *yawn_eye_dataset_new*[1], que possui $N = 2900$ registros distribuídos em $N = 4$ variáveis. O conjunto contém uma pasta *dataset_new* com duas subpastas, *train* e *test*, cada uma contendo 4 pastas com imagens relacionadas à fadiga e à sua ausência: *Closed* (olhos fechados), *Open* (olhos abertos), *yawn* (bocejando) e *no_yawn* (não bocejando).

[1] ⟨https://www.kaggle.com/datasets/serenaraju/yawn-eye-dataset-new/data⟩

Este trabalho foi desenvolvido utilizando dois modelos de rede neural, uma CNN simples e a *MobileNet*, visando comparar qual deles é mais eficiente para a detecção de fadiga no trânsito.

Pré-processamento dos dados

A etapa de pré-processamento seguiu um pipeline composto por: redimensionamento, normalização, mapeamento das classes e balanceamento de classes. No redimensionamento, as imagens foram ajustadas para uma resolução de 224x224 pixels, compatível com a arquitetura da *MobileNet*. A normalização dos pixels foi realizada de forma diferente para cada modelo, utilizando escalonamento para o intervalo $[0, 1]$ na CNN simples e a função *preprocess_input* para a *MobileNet*. O mapeamento das classes transformou as quatro categorias originais em um problema de classificação binária, onde *Closed* e *yawn* foram rotulados como 1 (fadiga) e *Open* e *no_yawn* como 0 (não fadiga). Para lidar com o desequilíbrio entre as classes, foi aplicada a técnica de pesos de classe, calculando pesos inversamente proporcionais à frequência das classes no conjunto de treino, garantindo que o modelo desse a devida importância à classe minoritária durante o treinamento.

CNN e MobileNet

A arquitetura da CNN foi projetada de forma sequencial e mais simples. Ela consiste em duas camadas convolucionais (*Conv2D*) seguidas por camadas de *Max Pooling* para redução de dimensionalidade. A primeira camada utiliza 32 filtros e a segunda 64 filtros, esta última aplicando regularização L2 para evitar *overfitting*. Após o achatamento dos dados, o modelo termina em uma camada densa de saída com ativação *Sigmoid*. A Figura 1 apresenta a representação gráfica da arquitetura da CNN simples, destacando as camadas e suas conexões.

O modelo *MobileNet* foi escolhido por ser uma arquitetura de baixo custo computacional e alta eficiência, ideal para aplicações em tempo real. Para adaptar a arquitetura do *MobileNet* ao problema específico de detecção de sonolência, foi desenvolvida uma arquitetura customizada utilizando a técnica de *Transfer Learning*.

O modelo base foi instanciado com pesos pré-treinados do *ImageNet*, omitindo as camadas densas superiores (*include_top=False*), com *global average pooling* aplicado às saídas convolucionais, e todas as suas camadas são congeladas para preservar os pesos pré-treinados. Sobre essa base, foi construído um classificador personalizado, utilizando a *Data Augmentation*, dessa forma as camadas de aumento de dados foram integradas diretamente ao modelo para melhorar a robustez, incluindo inversão horizontal (*RandomFlip*), rotação de até 10% (*RandomRotation*) e zoom de até 10% (*RandomZoom*).

A arquitetura da rede para este modelo adiciona duas camadas densas: a primeira com 256 neurônios e ativação *ReLU*, seguida por normalização em lote (*BatchNormalization*) e *Dropout* de 30%, enquanto a

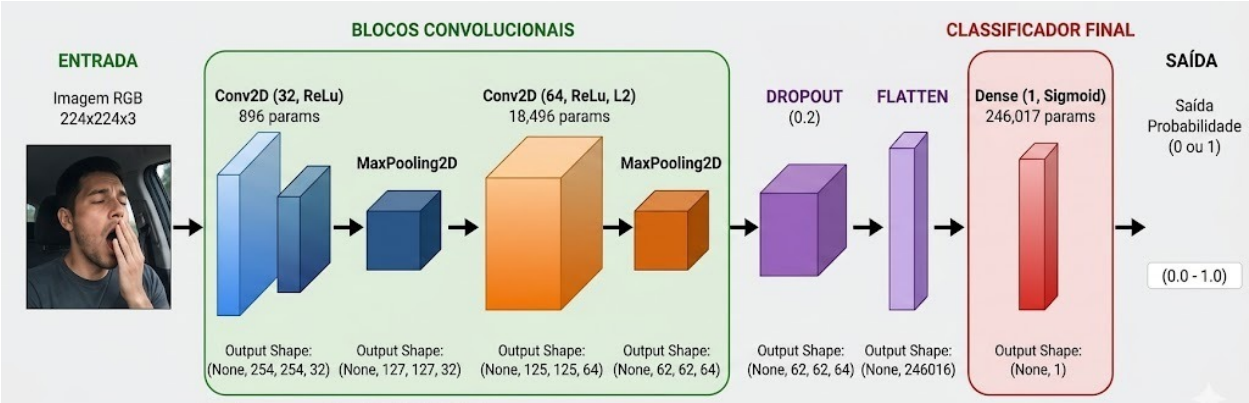


Figura 1: Representação da arquitetura da rede CNN Simples.

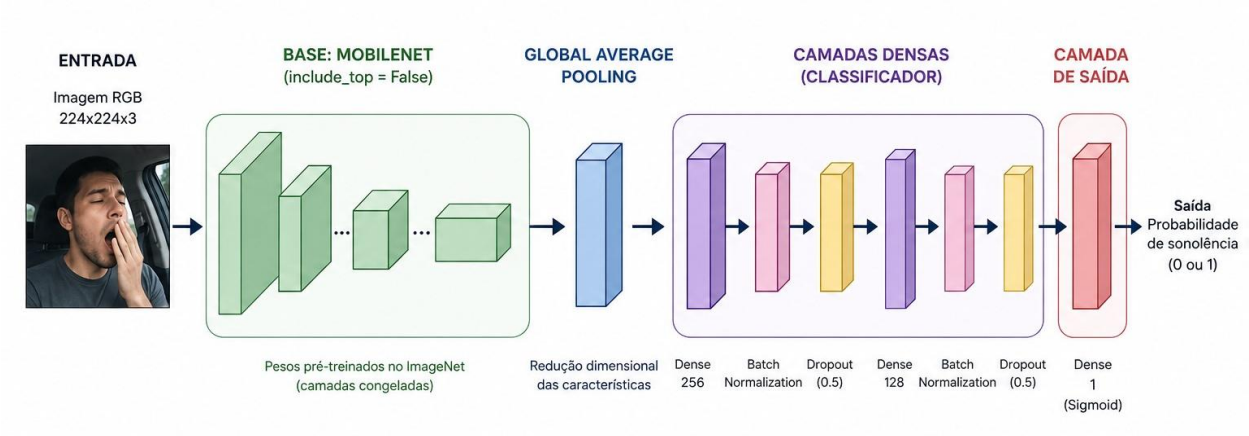


Figura 2: Representação da arquitetura da rede MobileNet.

segunda camada densa, com 128 neurônios e ativação *ReLU*, é seguida por *Dropout* de 20%. O modelo termina em uma camada densa de saída com ativação *Sigmoid* para classificação binária. A Figura 2 apresenta a representação gráfica da arquitetura da rede *MobileNet*.

Para o treinamento dos modelos o tamanho do lote (*batch size*) foi fixado em 32 amostras para ambos os modelos. O processo de treinamento foi definido com um limite máximo de 100 épocas, sendo monitorado pela técnica de para antecipada.

Dado que o problema foi mapeado como uma classificação binária (presença ou ausência de fadiga), a definição para otimização e avaliação utilizada foi a Entropia Cruzada Binária (*Binary Crossentropy*) como função de perda, visto que esta é ideal para medir a discrepância entre as probabilidades previstas e as classes reais em problemas de classificação binária. O ajuste dos pesos da rede foi realizado por meio do algoritmo *Adam*, selecionado pela sua eficiência computacional e capacidade de ajuste adaptativo da taxa de aprendizado. Complementarmente, a acurácia (*accuracy*) serviu como a métrica principal de avaliação, sendo monitorada durante todo o processo de treinamento para mensurar o percentual de predições corretas alcançadas pelos modelos.

Para evitar o *overfitting* e otimizar a convergência, foram implementadas algumas estratégias de regularização e controle. Inicialmente, o *Early Stopping* foi configurado para interromper o treinamento caso a perda de validação (*val_loss*) não apresentasse melhoria por 15 épocas consecutivas para a CNN, enquanto para a *MobileNet* foi utilizada uma paciência de 8 épocas, garantindo, ao final, a restauração automática dos pesos que obtiveram o melhor desempenho (*restore_best_weights=True*). Além disso, especificamente para o modelo MobileNet, adicionou-se o *callback ReduceLROnPlateau*, que monitora a perda de validação e reduz a taxa de aprendizado em um fator de 0,2 caso não haja melhora por 3 épocas consecutivas, permitindo que o modelo refine seus pesos com passos menores ao se aproximar de um mínimo local.

Para lidar com o desequilíbrio entre as classes de fadiga e não fadiga, utilizou-se o cálculo de pesos de classe (*class weights*). Esses pesos são calculados de forma inversamente proporcional à frequência das classes no conjunto de treino, garantindo que o modelo não seja tendencioso para a classe que possui mais amostras. Dessa forma, a função perda com os pesos pode ser definida como:

$$L = -\frac{1}{N}\sum_{i=1}^{N} w_{y_i} \cdot [y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)] \quad (1)$$

onde N é o número total de amostras, y_i é o rótulo verdadeiro da amostra i, $\hat{y}_i$ é a probabilidade prevista pelo

modelo para a classe positiva, e w_{y_i} é o peso associado à classe do rótulo y_i. Este ajuste na função de perda assegura que o modelo dê a devida importância às classes minoritárias, melhorando a capacidade de generalização e a performance do modelo em cenários de desequilíbrio de classes, como é o caso da detecção de fadiga, onde a classe de interesse (fadiga) pode ser menos representada em comparação à classe de não fadiga.

RESULTADOS E DISCUSSÃO

As simulações foram realizadas utilizando a biblioteca Keras com TensorFlow, em um ambiente de desenvolvimento Python 3.11. A análise comparativa entre os dois modelos revela que ambos apresentam um bom desempenho na classificação da fadiga, tendo uma sutil diferença, em que a *MobileNet* se apresenta melhor.

A Figura 3 apresenta a curva de perda da CNN, enquanto a Figura 4 mostra a curva de perda da *MobileNet*. Na curva perda modelo CNN simples, apresentada na (Figura 3), as oscilações entre a curva roxa que representa a validação e a curva azul que representa a perda, após as primeiras épocas, demonstram que a arquitetura tendeu a memorizar o padrão de treino. Embora isso tenha acontecido, o modelo apresenta métricas robustas no conjunto de teste.

Representada na (Figura 4), a curva *Loss* do modelo *MobileNet* apresenta uma convergência suave. A aproximação entre as curvas vermelha que representa o *Loss* e a curva roxa que representa a validação demonstra que o modelo conseguiu aprender evitando o risco de *overfitting*.

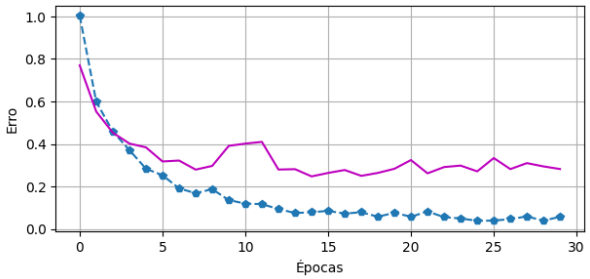


Figura 3: Curva de erro da CNN.

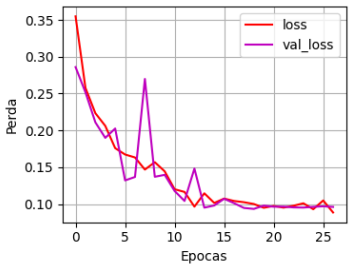


Figura 4: Curva de erro da *MobileNet*.

A Tabela 1 apresenta uma comparação das métricas de desempenho entre a CNN simples e a *MobileNet* no conjunto de teste. Ambos os modelos demonstram bom desempenho, com a CNN alcançando uma acurácia de 0,93, enquanto a *MobileNet* atingiu 0,95. A arquitetura *MobileNet* apresentou valores superiores em precisão, *Recall*, *F1-Score* e acurácia quando comparada à CNN Simples. O aumento no *Recall* demonstra que a *MobileNet* reduziu a ocorrência de falsos negativos. Além disso, os resultados sugerem que a utilização de uma arquitetura pré-treinada favoreceu a extração de características mais representativas das imagens, contribuindo para um desempenho mais consistente no conjunto de teste.

Tabela 1: Comparação das métricas de desempenho entre CNN Simples e *MobileNet* no conjunto de teste.

Modelo	Precisão	Recall	F1-Score	Acurácia
CNN	0,94	0,93	0,93	0,93
MobileNet	0,95	0,95	0,95	0,95

A matriz de confusão da CNN simples apresentada na Figura 5 mostra que, das 218 instâncias da classe não fadiga, o modelo classificou corretamente 202, cometendo apenas 16 erros na predição. Já da classe fadiga, o modelo conseguiu acertar 202 instâncias como verdadeiro positivo, classificando erroneamente 13 casos.

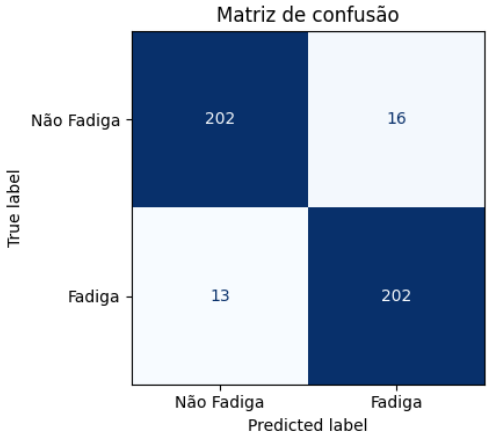


Figura 5: Matriz de Confusão da CNN simples.

Em contrapartida, o modelo baseado na arquitetura da *MobileNet*, ilustrada na Figura 6, apresenta uma predição de 205 instâncias corretas e 13 incorretas para a classe não fadiga. Já na classe fadiga, ele consegue um desempenho melhor ao conseguir classificar 205 instâncias corretas, reduzindo os falsos negativos para 10.

Para a segurança do motorista, a redução dos falsos negativos representa um ganho relevante, pois um falso negativo, nesse contexto, corresponde a um motorista em estado de fadiga que o sistema classifica erroneamente como não fadiga, o que nessas circunstâncias colocaria em risco a vida do motorista. O modelo *MobileNet* conseguiu reduzir de 13 para 10 em relação à CNN simples, o que, em um sistema real, pode representar uma diferença significativa. Em uma análise comparativa, o modelo da *MobileNet* apresenta melhor desempenho ao

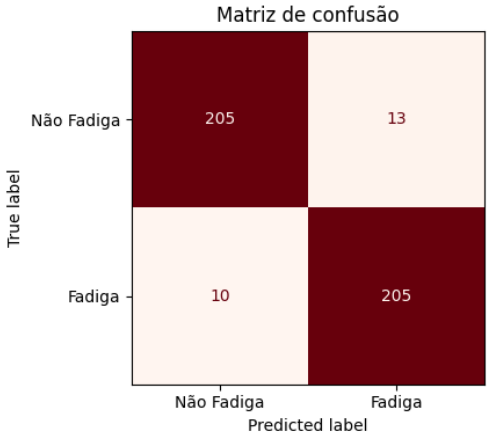


Figura 6: Matriz de Confusão da *MobileNet*.

entregar precisões estatísticas maiores, evitar que a rede decore os dados de treino e conseguir elevar a classe de acertos reduzindo o número de falsos negativos na matriz de confusão, mostrando-se, assim, mais adequado e confiável para aplicações de detecção de fadiga.

No modelo que obteve o melhor resultado foi incorporado o Grad-CAM (*Gradient-weighted Class Activation Mapping*) com o objetivo de visualizar e entender as decisões da rede. O Grad-CAM utiliza os gradientes da última camada convolucional para gerar mapas de calor que destacam as regiões mais importantes da imagem para a tomada de decisão do modelo.

Na Figura 7 pode-se observar que o modelo classificou corretamente as amostras. Na primeira linha, o mapa de calor do Grad-CAM concentrou-se na região da boca do condutor, que está bocejando, prevendo corretamente a classe sonolento, com 78,5% de probabilidade. Já na segunda linha, temos um olho fechado e o modelo fez a previsão com 100% de confiança, indicando a aprendizagem do modelo sobre essa característica importante para a detecção de fadiga. Por fim, na terceira linha, onde o motorista está desperto, o modelo previu com 100% de precisão a classe não sonolento, direcionando o mapa de calor para a região central do rosto, onde os olhos estão abertos, assim, validando a capacidade do modelo de distinguir o estado de sonolência e alerta.

CONCLUSÃO

Este trabalho investigou a aplicação de Redes Neurais Convolucionais, comparando dois modelos distintos, uma CNN simples desenvolvida e uma *MobileNet* por meio de *Transfer Learning*. Ambos os modelos foram treinados usando o mesmo *dataset* para a classificação binária de fadiga e não fadiga.

Os resultados obtidos demonstraram que a *MobileNet*, com 95% de acurácia no conjunto de teste, superou a CNN simples que atingiu 93% de acurácia. O desempenho superior da *MobileNet* pode se dar pelo uso de pesos pré-treinados no *ImageNet* aliado à aplicação do *Data Augmentation* e às técnicas de regularizações adotadas.

Apesar dos dados satisfatórios, vale ressaltar algumas limitações, dentre elas estão: o conjunto de dados utilizado é relativamente pequeno e foi coletado em um ambiente controlado, podendo não representar totalmente o mundo real. O modelo foi testado em ambiente *off-line*, sem a validação em tempo real.

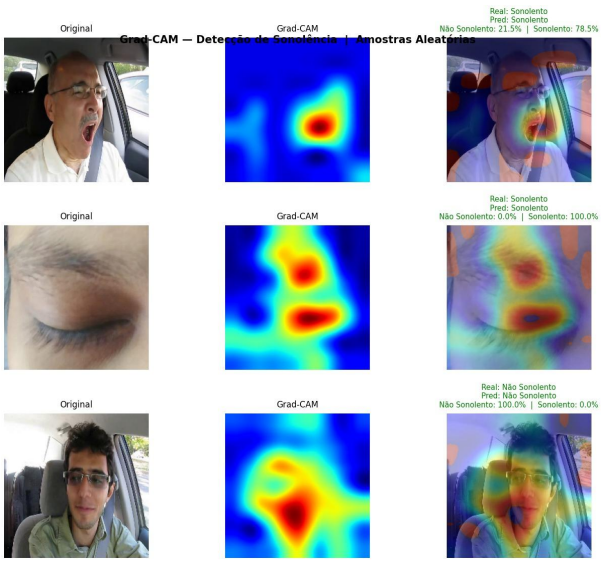


Figura 7: Grad-CAM.

Diante disso, para trabalhos futuros, visa-se a ampliação do conjunto de dados com amostras coletadas em condições reais de direção, considerando diferentes horários, tipos de iluminação e perfis de motoristas. Também é proposta a implementação do sistema em plataforma embarcada para validação em tempo real, permitindo avaliar o desempenho do modelo sob restrições computacionais. Além disso, outras arquiteturas de redes neurais, como *EfficientNet* e *ResNet*, podem ser investigadas em comparação com a *MobileNet*. Por fim, os resultados obtidos contribuem como base inicial promissora para o desenvolvimento de sistemas preventivos de acidentes causados por sonolência e fadiga no trânsito.

DECLARAÇÃO DE USO DE INTELIGÊNCIA ARTIFICIAL

Os autores declaram que utilizaram ferramentas de inteligência artificial, incluindo o Gemini, exclusivamente para auxiliar na melhoria da linguagem e legibilidade deste manuscrito. Após a utilização desta ferramenta, os autores revisaram e editaram o conteúdo conforme necessário e assumem total responsabilidade pelo conteúdo final do trabalho.

REFERÊNCIAS

CANANI, S. F.; BARRETO, S. S. M. Sonolência e acidentes automobilísticos. *Jornal de Pneumologia*, SciELO Brasil, v. 27, p. 94–96, 2001.

CHITTETI, C.; SRAVIKA, M.; AYESHA, S.; NAIK, D. S.; KUMAR, V. A.; KUMAR, M. P. Real-time driver drowsiness detection using mobile net for road safety enhancement. In: *2025 3rd International Conference on*

Disruptive Technologies (ICDT). [S.l.: s.n.], 2025. p. 692–697.

COSTA, L. C. d. et al. Monitoramento inteligente de fadiga: detecção de sonolência em motoristas com ia e visão computacional para aumentar a segurança no trânsito. 2024.

GIAP, Y. C.; MULJONO; AFFANDY; BASUKI, R. S.; AZIES, H. A.; ISNANTO, R. R.; DEWI, D. A. Attention-guided lightweight mobilenetv2 for real-time driver drowsiness classification on edge-iot systems. *International Journal of Advanced Computer Science and Applications*, The Science and Information Organization, v. 17, n. 1, 2026. Disponível em: ⟨http://dx.doi.org/10.14569/IJACSA.2026.0170142⟩.

OTTO, M.; WILL, J. C.; HOMANN, H. Ultra-lightweight eye state detection for resource-constrained devices. In: *2025 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. [S.l.: s.n.], 2025. p. 835–840.

RAJPUT, I. S.; TYAGI, S.; PANDEY, V.; UPRETI, A.; KUKREJA, J. K. Enhancing driver safety with mobilenetv2-based transfer learning for drowsiness detection. *Procedia Computer Science*, v. 259, p. 1239–1248, 2025. ISSN 1877-0509. Sixth International Conference on Futuristic Trends in Networks and Computing Technologies (FTNCT06), held in Uttarakhand, India. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S1877050925011809⟩.

SANTHIYA, S.; VIJAYAKANTH, M.; RAKSHETHA, R.; SANTHOSH, K.; TARRUN, S. et al. An intelligent vision-based system for driver fatigue detection using deep learning. In: IEEE. *2025 3rd International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT)*. [S.l.], 2025. v. 1, p. 310–315.

VIMALA, M.; NANDHINI, M.; JASMINE, S.; KUMAR, P. R.; RAMASAMY, S. Implementation of a lightweight deep learning model for detecting driver fatigue. *Franklin Open*, v. 15, p. 100547, 2026. ISSN 2773-1863. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S2773186326000630⟩.