Analysis of the Sustained Performance of Deep Learning Models for Driver Assistance in Mobile SoCs

Fabrício Daniel Freitas[1], **Arthur de Miranda Neto**[2], **Luiz Henrique Andrade Correa**[1]

[1]Departamento de Ciência da Computação (DCC) – Universidade Federal de Lavras (UFLA)
Caixa Postal 3037 – 37.200-000 – Lavras – MG – Brasil

[2]Departamento de Automática (DAT) – Universidade Federal de Lavras (UFLA)
Caixa Postal 3037 – 37.200-000 – Lavras – MG – Brasil

fabricio.freitas2@estudante.ufla.br, arthur.miranda@ufla.br, lcorreia@ufla.br

Abstract. *Advanced Driver Assistance Systems (ADAS) enhance road safety, and Edge AI enables its deployment on smartphones without dedicated hardware. This study evaluates the sustained performance of the SSD-MobileNet v1 deep learning model for passive ADAS on consumer devices, focusing on computing performance, energy efficiency, and thermal constraints. Using standard benchmarks and a custom Android application, we analyze three System-on-Chip (SoC) tiers: Snapdragon 8 Elite, Helio G99 Ultra, and Exynos 1330. Results show that entry and mid-range devices suffer mandatory CPU fallback due to driver immaturity, whereas flagships exhibit severe thermal throttling under sustained maximum loads.*

Resumo. *Sistemas Avançados de Assistência ao Condutor (ADAS) aumentam a segurança viária, e a Inteligência Artificial de Borda (Edge AI) viabiliza sua implementação em smartphones sem hardware dedicado. Este estudo avalia o desempenho sustentado do modelo de deep learning SSD-MobileNet v1 para ADAS passivo em dispositivos comerciais, focando em desempenho computacional, eficiência energética e restrições térmicas. Utilizando benchmarks padronizados e uma aplicação Android customizada, analisou-se três níveis de Systems-on-Chip (SoCs): Snapdragon 8 Elite, Helio G99 Ultra e Exynos 1330. Os resultados revelam que dispositivos de entrada e intermediários sofrem fallback compulsório para CPU devido à imaturidade de drivers, enquanto plataformas flagship enfrentam severo throttling térmico sob carga contínua.*

1. Introdução

Os sinistros de trânsito representam um desafio global urgente; no Brasil, causam mais de 50 mil fatalidades anuais [Carvalho 2023, Agência Nacional de Transportes Terrestres 2023]. Dados do Centro de Documentação e Memória (CEDOM) indicam que a desatenção e as reações tardias respondem por 42% dos acidentes no país, com impacto econômico superior a 50 bilhões de reais ao ano [CEDOM - Centro de Documentação e Memória do Mercado Segurador 2025]. Sistemas Avançados de Assistência ao Condutor (*Advanced Driver Assistance Systems* - ADAS) mitigam esse panorama com recursos ativos, a exemplo dos alertas de mudança de faixa [Renault 2024]. Contudo, o alto custo do hardware de fábrica restringe essa proteção a veículos de alto padrão [Weig 2016].

Para democratizar a segurança preventiva, a Inteligência Artificial de Borda (*Edge AI*) em smartphones permite executar algoritmos de *Deep Learning* localmente, usando sensores nativos e co-processadores dedicados chamados Unidades de Processamento Neural (*Neural Processing Units* - NPUs) [Badidi 2022, Chen et al. 2018, Ignatov et al. 2019]. Essa abordagem elimina a dependência exclusiva da nuvem e garante a baixa latência exigida em tempo real [Badidi 2022]. Entretanto, a inferência contínua impõe desafios físicos: a dissipação passiva provoca *throttling* térmico, reduzindo a frequência do *System-on-Chip* (SoC) e degradando a taxa de Quadros por Segundo (*Frames Per Second* - FPS) [Ignatov et al. 2021]. A fragmentação do ecossistema Android acentua o problema devido a discrepâncias em drivers e frameworks entre chipsets [Ignatov et al. 2019].

Este trabalho apresenta uma avaliação experimental do desempenho sustentado do modelo SSD-MobileNet v1 em tarefas ADAS passivas (Nível 1 SAE J3016) em smartphones comerciais processando vídeo em tempo real. O escopo da avaliação recai estritamente sobre a infraestrutura de hardware sob carga contínua extrema (*worst-case scenario*), isolando os limites físicos do silício de variáveis cognitivas do modelo de IA (como variações climáticas de chuva ou neblina). A mensagem central deste estudo é que a viabilidade prática da IA na borda móvel automotiva enfrenta um duplo gargalo: as severas restrições físicas de dissipação térmica do hardware sob estresse contínuo e a imaturidade do ecossistema de software (drivers e *frameworks*) para explorar todo o potencial do silício. Como evidência, demonstra-se que os SoCs de entrada e intermediários sofrem com o *fallback* compulsório para a Unidade Central de Processamento (CPU), enquanto a plataforma *flagship* enfrenta severo estrangulamento térmico. As contribuições centrais incluem: (i) a caracterização multidimensional comparativa de três SoCs sob estresse contínuo por 94 minutos; (ii) a identificação do impacto lógico de atualizações de sistema operacional sobre pipelines neurais; e (iii) a desmistificação estatística do volume bruto de detecções isoladas como métrica absoluta de desempenho.

O restante deste trabalho está organizado da seguinte forma: a Seção 2 apresenta a fundamentação teórica; a Seção 3 detalha a arquitetura lógica do sistema desenvolvido; a Seção 4 descreve a metodologia e o protocolo experimental; a Seção 5 expõe e analisa os resultados computacionais e físicos obtidos; e a Seção 6 consolida as conclusões do estudo.

2. Fundamentação Teórica

2.1. Sistemas Avançados de Assistência ao Condutor (ADAS)

Os Sistemas ADAS visam reduzir sinistros e otimizar a condução veicular [Masello et al. 2022]. A norma SAE J3016 estabelece seis níveis de automação, do Nível 0 (manual) ao 5 (pleno) [SAE International 2021]. Este estudo delimita-se ao Nível 1 (ADAS passivo ou informativo), operando como co-piloto digital [SAE International 2021]: o smartphone captura imagens pelo sensor óptico nativo, processa os dados localmente e emite alertas visuais ou sonoros, sem intervir diretamente sobre os atuadores mecânicos do veículo.

2.2. Inteligência Artificial de Borda (Edge AI)

A *Edge AI* executa modelos de *Deep Learning* em dispositivos terminais, descentralizando o pipeline em nuvem [Badidi 2022, Li et al. 2022]. No setor automotivo, essa

abordagem é imperativa por razões operacionais [Chen et al. 2018]. O processamento local elimina a dependência de redes celulares sujeitas a zonas de sombra, assegurando operação contínua [Musa et al. 2022]. Adicionalmente, mitiga a latência de rede, garantindo respostas em milissegundos para alertas em tempo real e preservando a privacidade ao restringir as imagens ao próprio aparelho [Katare et al. 2023, Badidi 2022].

2.3. Hardware e Evolução de SoCs Móveis

A inferência local é viabilizada pela evolução dos SoCs para arquiteturas altamente heterogêneas [Ignatov et al. 2019]. Esses chipsets integram CPUs, Unidades de Processamento Gráfico (*Graphics Processing Units* - GPUs) paralelas e aceleradores dedicados (NPUs ou APUs), que utilizam matrizes de processadores de fluxo para computar produtos escalares e convoluções com alta eficiência energética [Ignatov et al. 2019, Ignatov et al. 2021, Lin et al. 2021]. Avaliam-se três SoCs representativos das restrições atuais do mercado: o Snapdragon 8 Elite (Premium), com núcleos Oryon e NPU Hexagon nativamente mapeável via QNN SDK; o Helio G99 Ultra (Intermediário), com núcleos Cortex-A76/A55, GPU Mali-G57 MC2 e APU de arquitetura fechada; e o Exynos 1330 (Entrada), com núcleos Cortex-A78/A55 e GPU Mali-G68 MP2, desprovido de um acelerador neural com suporte aberto para bibliotecas genéricas de terceiros.

2.4. Frameworks de Inferência e Técnicas de Otimização

Frameworks como o *TensorFlow Lite* (TFLite) mapeiam os grafos computacionais dos modelos para os aceleradores físicos do SoC. No ecossistema móvel, esse acesso aos drivers é padronizado primariamente via *delegates* da *Android Neural Networks API* (NNAPI), que atua como uma camada de abstração genérica independente do fabricante para democratizar a aceleração em hardware [Wulfert et al. 2024, Elhanashi et al. 2024]. Para contornar o alto custo computacional de redes em ponto flutuante de 32 bits (FP32), aplica-se a quantização pós-treinamento (*Post-Training Quantization* - PTQ) [Jacob et al. 2018]. O método reduz a precisão de pesos e ativações para inteiros de 8 bits (INT8) via escala linear uniforme:

$$q = \text{round}\left(\frac{x}{\text{scale}} + \text{zero_point}\right) \tag{1}$$

onde q é o valor quantizado, x é o dado em FP32, scale determina a divisão do intervalo e zero_point atua como deslocamento de calibração [Jacob et al. 2018]. A quantização INT8 diminui o uso de memória em até 75% com perdas marginais de precisão [Jacob et al. 2018], enquanto a poda (*pruning*) elimina conexões com pesos irrelevantes, reduzindo as operações flutuantes (*Floating Point Operations* - FLOPs) necessárias [Han et al. 2015].

2.5. Arquitetura SSD-MobileNet v1

O modelo SSD-MobileNet v1 integra o detector *Single Shot Detector* (SSD) ao *backbone* MobileNet v1 para viabilizar detecções leves e estáveis [Wang et al. 2025, Chen et al. 2018]. A escolha desta arquitetura clássica, em detrimento de detectores mais recentes (como YOLOv8n ou EfficientDet-Lite), justifica-se pela sua robustez e maturidade no ecossistema Android para inferência quantizada em inteiros (INT8). Modelos complexos frequentemente carecem de suporte universal nas camadas de abstração de

hardware (HAL) de dispositivos de entrada, o que forçaria um *fallback* imediato e inviabilizaria a avaliação isolada dos aceleradores neurais (NPUs/APUs). O SSD opera em estágio único, calculando simultaneamente as probabilidades de classe e as coordenadas das caixas delimitadoras (*bounding boxes*) em uma única passagem pelo grafo, o que mitiga a latência [Chen et al. 2018]. A eficiência advém das convoluções separáveis em profundidade (*depthwise separable convolutions*), que decompõem a operação convencional em uma filtragem espacial por canal (*depthwise*) e uma combinação pontual (*pointwise*) de 1×1 [Wang et al. 2025]. Essa abordagem reduz parâmetros e custo computacional por um fator proporcional a:

$$\frac{1}{N} + \frac{1}{D_k^2} \tag{2}$$

onde N é o número de canais de saída e D_k representa a dimensão espacial do filtro convolucional (tipicamente 3×3) [Wang et al. 2025], garantindo taxas de atualização fluidas em hardware móvel restrito.

3. Desenvolvimento da Aplicação e Coleta de Dados

Para viabilizar a validação prática dos modelos em ambiente de borda, foi desenvolvida uma aplicação móvel ADAS customizada em ambiente nativo Android utilizando as linguagens Java e Kotlin. O núcleo computacional do sistema foi estruturado ao redor do *runtime* do TFLite, que atua como o motor de inferência responsável pelo carregamento e execução do grafo quantizado da rede SSD-MobileNet v1. O pipeline de software gerencia o fluxo de quadros provenientes do sensor óptico, realiza o pré-processamento das matrizes de pixels para as dimensões de entrada do modelo e despacha os blocos aritméticos para os subcomponentes de aceleração via camadas de delegação física (*delegates*).

Com o intuito de registrar o impacto de sessões prolongadas de inferência sem introduzir vieses ou gargalos de concorrência no processamento visual, implementou-se um mecanismo assíncrono de monitoramento operando em segundo plano (*background service*). Esse módulo de telemetria realiza consultas diretas ao sistema de arquivos virtuais do kernel Linux do ecossistema Android (*sysfs*), extraindo variáveis de estado em intervalos estritos de 1 segundo. Esse perfilamento contínuo (*profiling*) permite atestar fisicamente o nível de carga alocado nos núcleos da CPU em cenários de *fallback* da NNAPI, validando as latências medidas. Os algoritmos implementados consolidam métricas computacionais (taxa instantânea de FPS e latência da rede), térmicas (temperatura do silício e da bateria) e energéticas (corrente e potência em miliwatts). Ao final de cada ciclo, os dados brutos estruturados são serializados e descarregados localmente na memória Flash do dispositivo em arquivos de extensão CSV (*Comma-Separated Values*).

Para fomentar a transparência, a reprodutibilidade metodológica e o avanço das pesquisas em computação de borda, o código-fonte da aplicação móvel e os *scripts* de coleta foram disponibilizados publicamente em repositório aberto[1].

4. Metodologia Experimental

Esta seção detalha a abordagem empírica para avaliar a inferência de *Deep Learning* na borda móvel. O pipeline compreende a seleção de dispositivos, o uso de benchmarks sintéticos e o teste controlado da aplicação Android dedicada.

[1]Repositório do projeto: `https://github.com/fabricio-ifmg/adas-research-app.git`

4.1. Plataformas de Hardware e Dispositivos Selecionados

Para mapear a execução de redes neurais em diferentes segmentos de mercado e capacidades de hardware, selecionaram-se três smartphones representativos dos segmentos Premium, Intermediário e Entrada, com especificações consolidadas na Tabela 1.

Tabela 1. Especificações técnicas detalhadas das plataformas de teste

Especificação	Galaxy S25	Redmi Note 14 4G	Galaxy A14 5G
Segmento	Premium	Intermediário	Entrada
SoC	Snapdragon 8 Elite	MediaTek Helio G99 Ultra	Exynos 1330
Litografia	3 nm (TSMC N3E)	6 nm (TSMC)	5 nm (Samsung)
CPU	2x Oryon V2 L (4,47 GHz) 6x Oryon V2 M (3,53 GHz)	2x Cortex-A76 (2,2 GHz) 6x Cortex-A55 (2,0 GHz)	2x Cortex-A78 (2,4 GHz) 6x Cortex-A55 (2,0 GHz)
GPU	Adreno 830	Mali-G57 MC2	Mali-G68 MP2
Acelerador IA	Hexagon / QNN HTP	APU Integrada	Não possui NPU dedicada
Memória RAM	12 GB LPDDR5X	8 GB LPDDR4X	4 GB LPDDR4X
Sistema / ROM	Android 15 (One UI 7)	Android 15 (HyperOS)	Android 14 / Android 15

4.2. Ferramentas de Avaliação e Inferência

A infraestrutura de avaliação combinou benchmarks sintéticos e a aplicação prática. O AI Benchmark (v6.0.2) mapeou o pico de desempenho computacional de IA, contrastando a execução isolada na CPU com a aceleração dedicada em hardware (NN). Os testes empregaram rotinas locais baseadas nas arquiteturas MobileNet-V3 [Howard et al. 2019], EfficientNet-B4 [Tan and Le 2019] e DeepLab-V3+ [Chen et al. 2018], consolidando os índices de velocidade e exatidão [Ignatov et al. 2019]. Em paralelo, o Burnout Benchmark (v2.0.8) avaliou a estabilidade sob estresse severo ao sobrecarregar simultaneamente CPU, GPU e aceleradores do SoC, monitorando a queda de FPS, a retenção de performance e a dissipação térmica passiva ao longo do tempo [Ignatov et al. 2021].

4.3. Configuração e Protocolo Experimental

Os testes ocorreram em ambiente controlado a 21°C. Essa temperatura foi fixada metodologicamente para isolar o orçamento térmico (*thermal budget*) nominal do silício, evitando que variações climáticas externas antecipassem o *throttling*. Os aparelhos operaram desconectados da rede elétrica, com as interfaces de rede sem fio (Wi-Fi e Bluetooth) desativadas e com bateria mantida acima de 50%, medidas adotadas para suprimir gargalos de processamento em segundo plano e impedir o acionamento de modos automáticos de economia de energia impostos pelo sistema operacional.

O protocolo físico dividiu-se em dois regimes: o **Regime de Dissipação de Pico (Benchmarks)** e o **Regime de Aplicação ADAS**. No segundo regime, o estímulo visual foi gerado pela exibição de vídeos de trajetos reais em uma tela OLED 4K a 60 FPS alinhada ao campo óptico das câmeras, simulando a perspectiva do para-brisa. Utilizaram-se propositalmente vídeos com condições de visibilidade adequadas, isolando os gargalos físicos e térmicos do *hardware* de possíveis vieses cognitivos da rede neural frente a intempéries severas (como chuva ou neblina) [Oz 2023, Utah 2023].

A carga contínua totalizou 94 minutos de processamento visual ininterrupto, combinando um percurso noturno de 52 minutos [Utah 2023] e um diurno de 41 minutos

[Oz 2023]. A ausência de mecanismos de inferência adaptativa para poupar bateria (impondo um *duty cycle* de 100%) foi delineada para estressar o limite físico sustentado de forma homogênea em todos os aparelhos. Para garantir robustez estatística e replicabilidade, realizaram-se oito execuções consecutivas completas por dispositivo e cenário operacional.

4.4. Tratamento de Dados e Métricas Coletadas

Os dados brutos coletados segundo a segundo pelas rotinas em segundo plano foram submetidos a um pipeline de tratamento estatístico. Para mitigar a influência de ruídos computacionais espúrios transitórios (como chamadas de rotinas cíclicas do sistema operacional Android), aplicou-se o método do Intervalo Interquartil (*Interquartile Range* - IQR) para detecção e eliminação de *outliers*. As métricas consolidadas finais representam a média aritmética acompanhada do cálculo de desvio padrão (σ) para evidenciar a variabilidade temporal dos resultados.

As variáveis monitoradas dividiram-se em quatro escopos centrais: **Métricas Computacionais** (tempo de inferência em ms, taxa de quadros instantânea e média); **Métricas Físicas e Térmicas** (temperaturas do SoC e da bateria em °C, e delta de elevação térmica $\Delta T = T_{\text{final}} - T_{\text{inicial}}$); **Métricas Energéticas** (potência instantânea em W e taxa percentual de descarga da bateria); e **Métricas de Visão Computacional** (mapeamento quantitativo bruto de objetos detectados e distribuição acumulada por classes de interesse, como carros, caminhões, motocicletas, pedestres e semáforos).

4.5. Ameaças à Validade

Com o intuito de assegurar a transparência científica do estudo, mapearam-se quatro categorias de ameaças à validade:

- **Ameaças Internas:** Relacionam-se à instabilidade de frequências induzida pelas políticas dinâmicas de escalonamento térmico em nível de kernel (*DVFS thermal governor*) nativas de cada fabricante e à ocorrência de atualizações de *firmware* durante o período experimental.
- **Ameaças Externas:** A amostragem restrita a três chipsets específicos impede a generalização irrestrita dos achados para todo o ecossistema móvel. Adicionalmente, a simulação visual baseada em telas OLED em ambiente controlado desconsidera variáveis do mundo real, como trepidação mecânica severa e radiação solar infravermelha direta sobre o chassi do aparelho.
- **Ameaças de Construção:** O foco delimitou-se à infraestrutura computacional e aos limites físicos do hardware sob estresse (latência, temperatura e potência). A decisão de não empregar métricas clássicas de qualidade de visão computacional com gabarito (*ground truth*), como mAP, precisão e revocação, restringe as conclusões à sustentabilidade operacional da execução do grafo, delegando a validação da acurácia semântica a trabalhos futuros.
- **Ameaças de Conclusão:** A contagem volumétrica bruta de marcações introduz distorções interpretativas decorrentes do paradoxo da taxa de quadros. Ressalta-se que, embora o *throttling* térmico não altere a precisão matemática dos pesos estáticos do modelo, a severa degradação de FPS resultante afeta drasticamente a eficácia prática da detecção ao reduzir a resolução temporal do sistema e induzir artefatos físicos de dessincronização na captura do sensor.

5. Resultados e Discussão

Esta seção apresenta os dados experimentais obtidos sob os regimes propostos, dividindo-se entre a caracterização de pico sintético, a resiliência térmica sustentada e o desempenho fim a fim na aplicação ADAS.

5.1. Avaliação de Pico Computacional (AI Benchmark)

Os ensaios com o *AI Benchmark* mapearam o limite teórico superior dos SoCs, evidenciando o impacto de drivers e *delegates*. A Tabela 2 consolida os tempos médios de inferência obtidos para as arquiteturas MobileNet-V3, EfficientNet-B4 e DeepLab-V3+, expressos na notação **CPU / NN** (tempo em milissegundos na CPU central versus tempo sob aceleração de hardware por co-processador neurais/gráficos).

Tabela 2. Tempos médios de inferência (ms) do tipo CPU / NN no AI Benchmark

Modelo e Precisão Computacional	Galaxy S25	Redmi Note 14 4G	Galaxy A14 ($5G^1$)	Galaxy A14 ($5G^2$)
MobileNet-V3 INT8 ($\sigma \pm 1,2$)	22,5 / 0,63	82,4 / 483,0	58,6 / 34,8	58,9 / 35,0
MobileNet-V3 FP16 ($\sigma \pm 2,4$)	60,4 / 8,76	133,0 / 51,9	117,0 / 33,7	119,0 / 34,0
EfficientNet-B4 INT8 ($\sigma \pm 5,1$)	48,6 / 1,61	271,0 / 2434,0	411,0 / 198,0	331,0 / 131,0
EfficientNet-B4 FP16 ($\sigma \pm 8,7$)	103,0 / 26,0	479,0 / 175,0	625,0 / 538,0	513,0 / 116,0
DeepLab-V3+ INT8 ($\sigma \pm 12,3$)	106,0 / 2,00	557,0 / 4960,0	617,0 / 670,0	437,0 / 404,0
DeepLab-V3+ FP16 ($\sigma \pm 21,0$)	232,0 / 31,5	1021,0 / 499,0	1239,0 / 1728,0	1127,0 / 391,0

[1] Dispositivo avaliado sob o sistema Android 14 (ROM One UI 6).
[2] Dispositivo avaliado após a atualização para o sistema Android 15 (ROM One UI 7).

Os dados revelam expressiva vantagem para o Galaxy S25 (Premium). Apoiado pelo *delegate* proprietário Qualcomm Neural Network (QNN HTP), o Snapdragon 8 Elite registrou latências na escala de submilissegundos em quantização inteira ($0,63$ ms na MobileNet-V3 INT8), validando a viabilidade de visão computacional concorrente em hardware de última geração.

Em contrapartida, o Redmi Note 14 4G (Helio G99 Ultra) demonstrou falhas de integração de *software*. A ativação da aceleração via NNAPI elevou a latência em INT8 de $82,4$ ms (CPU) para $483,0$ ms no modelo leve, atingindo $4960,0$ ms no DeepLab-V3+. Esse *fallback* compulsório ocorre porque o framework TFLite falha ao mapear operadores inteiros para a APU no ecossistema HyperOS, gerando um *overhead* de serialização que onera a CPU central. O dispositivo apresentou melhor eficiência em FP16 ($51,9$ ms), indicando que a GPU Mali-G57 obteve mapeamento de ponto flutuante via NNAPI mais maduro que o acelerador de IA de arquitetura fechada.

No Galaxy A14 5G (Exynos 1330), as métricas evidenciaram o impacto das atualizações do sistema operacional. Sob o Android 14 (A14 5G[1]), o aparelho sofreu *fallback* sistemático para CPU em segmentação (NN-FP16 de $1728,0$ ms). Com o Android 15 e One UI 7 (A14 5G[2]), o pipeline foi otimizado, reduzindo a latência do DeepLab-V3+ em mais de 77% (para $391,0$ ms) e da EfficientNet-B4 para $116,0$ ms. Isso confirma que o limite de desempenho da inferência na borda móvel é fortemente condicionado pela evolução lógica das camadas de abstração do sistema operacional.

5.2. Sustentabilidade Térmica e de Estresse (Burnout Benchmark)

Os ensaios de estresse prolongado com o *Burnout Benchmark* avaliaram a sustentação do *throughput* sob dissipação puramente passiva. Sob carga contínua isolada de NPU

(FP16), o Galaxy S25 iniciou com 182,5 FPS, mas o adensamento térmico reduziu a taxa estável para 44,9 FPS (degradação superior a 75%) com a bateria a 42,6°C. No estresse combinado máximo (CPU + NPU), a elevação térmica superou as barreiras de tolerância do fabricante, forçando o sistema operacional a encerrar a aplicação para salvaguarda do hardware após a carcaça atingir 43,1°C.

O Redmi Note 14 4G apresentou estabilidade em sua curva operacional. Sob estresse de CPU e GPU, variou de 4,5 FPS iniciais para 4,1 FPS sustentados, limitando a temperatura máxima a 40,0°C. Esse perfil denota uma política de gerenciamento térmico (*DVFS thermal governor*) conservadora, que prioriza a resiliência de longo prazo em detrimento de picos efêmeros, sendo adequada para monitoramento veicular prolongado.

O Galaxy A14 5G evidenciou a influência das atualizações de *firmware*. Sob o Android 14 (One UI 6), o estresse de CPU elevou a temperatura interna para 45,7°C, com *throttling* severo que derrubou o rendimento de 5,3 para 2,8 FPS. Com o Android 15, a reestruturação do escalonamento dos cores Cortex otimizou a execução, que operou de 7,2 a 3,6 FPS, reduzindo a temperatura de pico para 32,5°C.

5.3. Avaliação de Desempenho Fim a Fim na Aplicação ADAS

Após a calibração, os aparelhos executaram os pipelines de vídeo diurno e noturno na aplicação SSD-MobileNet v1. A Tabela 3 compila as métricas de infraestrutura de hardware consolidadas pela média aritmética após as rodadas completas.

Tabela 3. Métricas consolidadas de desempenho prático na aplicação ADAS

Indicador Métrico Operacional	Galaxy S25	Redmi Note 14 4G	Galaxy A14 5G
Duração Efetiva do Teste (minutos)	96,6 min	96,9 min	96,5 min
Total de Quadros Processados	173.455	107.021	53.254
Tempo de Inferência Médio (ms)	**24,3** ($\pm 1,8$)	**57,3** ($\pm 3,9$)	**119,7** ($\pm 9,2$)
Tempo de Inferência Mín. / Máx. (ms)	12,0 / 61,0	37,0 / 151,0	33,0 / 374,0
Taxa de Quadros Média (FPS)	**29,9** ($\pm 0,4$)	**18,4** ($\pm 1,1$)	**9,2** ($\pm 1,9$)
Taxa de Quadros Mínima / Máxima	8,0 / 48,0	4,0 / 24,0	2,0 / 24,0
Consumo Total de Bateria (%)	39,0%	22,0%	33,0%
Taxa de Descarga por Minuto	0,40%/min	0,23%/min	0,34%/min
Temperatura Inicial / Final SoC (°C)	52,3 / 62,6 (ΔT +10,3)	N/A	N/A
Temperatura Inicial / Final Bat. (°C)	31,9 / 39,0 (ΔT +7,1)	23,0 / 36,0 (ΔT +13,0)	25,2 / 27,8 (ΔT +2,6)

Os dados confirmam o desempenho do Galaxy S25 (Premium), que sustentou uma taxa média de 29,9 FPS e processou mais de 173 mil quadros. Sua latência média de inferência situou-se em 24,3 ms, com picos máximos limitados a 61,0 ms. Sob a perspectiva computacional, esse comportamento atende com folga ao orçamento de tempo inferior a 100 ms exigido na literatura automotiva para a garantia de determinismo e segurança em alertas de tempo real [Katare et al. 2023]. Sob a ótica energética, contudo, o S25 registrou a maior demanda, consumindo 39,0% de bateria (0,40%/min).

O Redmi Note 14 4G atuou como uma plataforma intermediária equilibrada: com latência média de 57,3 ms, manteve 18,4 FPS estáveis e obteve a maior eficiência energética do estudo (22,0% de consumo total ou 0,23%/min). Já o Galaxy A14 5G (avaliado sob o sistema Android 15 / One UI 7) sinalizou limitações severas de infraestrutura computacional para tarefas ativas; sua latência média de 119,7 ms e quedas frequentes

para até 2 FPS geram uma descontinuidade temporal que restringe o dispositivo de entrada a análises estatísticas em cenários de menor criticidade temporal.

5.4. O Paradoxo das Detecções Brutas e Artefatos de Recontagem

A avaliação quantitativa do pipeline de visão computacional revelou um resultado contraintuitivo: o volume total de detecções brutas seguiu uma ordem inversa ao poder computacional dos aparelhos. Como sumarizado na Tabela 4, o modelo de entrada (Galaxy A14 5G) registrou o maior índice de marcações (21.930), seguido pelo intermediário (20.950) e pelo flagship Galaxy S25 (20.063).

Tabela 4. Distribuição acumulada de detecções por classe e dispositivo

Classe de Interesse ADAS	Galaxy S25	Redmi Note 14 4G	Galaxy A14 5G
Veículos (Carros)	14.301	14.846	17.234
Veículos Pesados (Caminhões)	52	34	39
Motocicletas	18	5	4
Elementos Urbanos (Pessoas)	5.274	4.497	2.231
Sinalização Eletrônica (Semáforos)	418	1.568	2.422
Total Acumulado de Detecções	**20.063**	**20.950**	**21.930**

Esse comportamento ilustra o paradoxo da taxa de quadros, evidenciando uma limitação metodológica da inferência baseada em quadros independentes (*frame-by-frame processing*) sem o suporte de algoritmos para rastreamento temporal (*multi-object tracking*). Como a aplicação analisa cada frame isoladamente, um mesmo objeto estático ou em deslocamento lento à frente acaba sendo computado repetidas vezes a cada novo ciclo de inferência.

Devido à sua baixa taxa de amostragem (9,2 FPS), o Galaxy A14 5G permanece mais tempo exposto ao mesmo estímulo visual estático entre as transições de cena, amplificando artificialmente a contagem final através de sucessivas recontagens do mesmo objeto em múltiplos *frames*. Em contrapartida, o Galaxy S25, operando em alta fluidez (29,9 FPS), captura com precisão a dinâmica temporal e as variações rápidas de movimentação e oclusão de cena. Assim, os resultados demonstram que, no domínio dos sistemas embarcados críticos, o volume bruto de detecções isoladas é uma métrica falaciosa, devendo a avaliação pautar-se estritamente na continuidade e na estabilidade da latência temporal.

5.5. Limitações e Trabalhos Futuros

A despeito da consistência dos dados de infraestrutura coletados, a principal limitação deste estudo reside na ausência de uma validação quantitativa da qualidade lógica das marcações da rede. Como o ambiente de teste não integrou um gabarito anotado em nível de pixel ou caixa delimitadora (*ground truth*), os índices de acurácia computacional, precisão, revocação e precisão média global (*mean Average Precision* - mAP) não puderam ser extraídos, impedindo a aferição de falsos positivos ou negativos causados por oclusões ou variações de iluminação noturna.

Para mitigar essas barreiras técnicas e expandir a robustez das soluções de ADAS móveis, recomendam-se as seguintes diretrizes para investigações futuras: (i) adotar pipelines com *tracking* temporal via algoritmos leves de rastreamento multiobjeto (*Multi-Object Tracking* - MOT) para correlacionar os vetores de detecção e erradicar artefatos de

recontagem cíclica; (ii) explorar SDKs nativos dos fabricantes (Qualcomm QNN, MediaTek NeuroPilot e Samsung Neural) em substituição direta à camada genérica de abstração da NNAPI para forçar o acionamento de NPUs e APUs em modelos INT8; (iii) implementar validação quantitativa baseada em gabaritos manuais terrestres para calcular matematicamente a acurácia de detecção nos cenários diurno e noturno; e (iv) realizar uma avaliação interplataforma estendendo o escopo experimental para o ecossistema Apple iOS via *runtime* LiteRT e CoreML delegate na *Apple Neural Engine*, quantificando a eficiência de plataformas verticalmente integradas e altamente homogêneas.

6. Conclusão

Este trabalho apresentou uma avaliação experimental multidimensional do desempenho sustentado de modelos de *Deep Learning* para ADAS passivos executados em smartphones comerciais. Ao confrontar as especificações nominais contra regimes contínuos de estresse térmico, energético e lógico de 94 minutos, caracterizaram-se os limites físicos reais da Inteligência Artificial de Borda na periferia móvel.

Os resultados evidenciaram que o abismo de desempenho entre os chipsets vai além da capacidade bruta do silício. O Galaxy S25 atendeu de forma estrita às restrições temporais de segurança veicular em tempo real, operando a estáveis 29,9 FPS, embora o estresse extremo tenha provocado o acionamento de travas térmicas do sistema operacional Android. O Redmi Note 14 4G destacou-se pela previsibilidade e alta eficiência energética, qualificando-se para telemetria prolongada de menor criticidade, enquanto o Galaxy A14 5G expôs as restrições severas do segmento de entrada devido à alta latência da CPU. O estudo desmistificou o volume de detecções brutas como indicador absoluto de desempenho, provando o paradoxo da recontagem cíclica induzida por baixas taxas de amostragem. Por fim, mapeou-se o gargalo crônico imposto pela imaturidade de drivers abertos no ecossistema Android, reforçando a necessidade de otimizações lógicas verticais para a consolidação de sistemas críticos embarcados na borda.

Referências

Agência Nacional de Transportes Terrestres (2023). Boletim anual de sinistros de trânsito em rodovias 2023. Technical report, ANTT, Superintendência de Infraestrutura Rodoviária, Brasília.

Badidi, E. (2022). Edge ai and blockchain for smart sustainable cities: Promise and potential. *Sustainability*, 14(13).

Carvalho, C. H. R. d. (2023). Balanço da 1ª década de ação pela segurança no trânsito no brasil e perspectivas para a 2ª década. Technical report, Instituto de Pesquisa Econômica Aplicada (Ipea), Brasília.

CEDOM - Centro de Documentação e Memória do Mercado Segurador (2025). Acidentes de trânsito no brasil: de 2006 a 2025, muitas mudanças. `https://www.editoraroncarati.com.br/v2/Artigos-e-Noticias/Artigos-e-Noticias/Acidentes-de-transito-no-Brasil-de-2006-a-2025-muitas-mudancas.html`. Acesso em: 01 jun. 2025.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In Ferrari, V.,

Hebert, M., Sminchisescu, C., and Weiss, Y., editors, *Computer Vision – ECCV 2018*, pages 833–851, Cham. Springer International Publishing.

Elhanashi, A., Dini, P., Saponara, S., and Zheng, Q. (2024). Advancements in tinyml: Applications, limitations, and impact on iot devices. *Electronics*, 13(17).

Han, S., Pool, J., Tran, J., and Dally, W. J. (2015). Learning both weights and connections for efficient neural networks. In *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1135–1143, Cambridge, MA, USA. MIT Press.

Howard, A., Sandler, M., Chen, B., Wang, W., Chen, L.-C., Tan, M., Chu, G., Vasudevan, V., Zhu, Y., Pang, R., Adam, H., and Le, Q. (2019). Searching for mobilenetv3. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1314–1324.

Ignatov, A., Malivenko, G., Timofte, R., Zhang, Z., Wang, Y., Huang, Z., Luo, G., and Yu, G. (2021). Fast and accurate single-image depth estimation on mobile devices, mobile ai 2021 challenge: Report. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2545–2557.

Ignatov, A., Timofte, R., Chou, W., Wang, K., Wu, M., Hartley, T., and Van Gool, L. (2019). AI Benchmark: All about deep learning on smartphones in 2019. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 3617–3635.

Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., and Kalenichenko, D. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2704–2713.

Katare, D., Perino, D., Nurmi, J., Warnier, M., Janssen, M., and Ding, A. Y. (2023). A survey on approximate edge ai for energy efficient autonomous driving services. *IEEE Communications Surveys & Tutorials*, 25(4):2714–2754.

Li, P., Wang, X., Huang, K., Huang, Y., Li, S., and Iqbal, M. (2022). Multi-model running latency optimization in an edge computing paradigm. *Sensors*, 22(16).

Lin, W., Adetomi, A., and Arslan, T. (2021). Low-power ultra-small edge ai accelerators for image recognition with convolution neural networks: Analysis and future directions. *Electronics*, 10(17).

Masello, L., Castignani, G., Sheehan, B., Murphy, F., and McDonnell, K. (2022). On the road safety benefits of advanced driver assistance systems in different driving contexts. *Transportation Research Interdisciplinary Perspectives*, 15:100670.

Musa, S. S., Zennaro, M., Libsie, M., and Pietrosemoli, E. (2022). Convergence of information-centric networks and edge intelligence for iov: Challenges and future directions. *Future Internet*, 14(7):192.

Oz, W. (2023). Berlin - germany 4k driving tour 2023 — potsdamer platz, charlottenburg, wilmersdorf, spandau. Vídeo no YouTube. Disponível em: `https://www.youtube.com`. Acesso em: 28 jun. 2025.

Renault (2024). Sistema de assistência ao condutor (adas) - programa human first. https://www.renault.com.br/programahumanfirst/adas.html. Acesso em: 23 ago. 2024.

SAE International (2021). Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. *SAE Technical Standard*, J3016_202104.

Tan, M. and Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, pages 6105–6114.

Utah, J. (2023). Chicago 4k - night drive - driving downtown. Vídeo no YouTube. Disponível em: https://www.youtube.com. Acesso em: 28 jun. 2025.

Wang, X., Tang, Z., Guo, J., Meng, T., Wang, C., Wang, T., and Jia, W. (2025). Empowering edge intelligence: A comprehensive survey on on-device ai models. *ACM Comput. Surv.*, 57(9).

Weig, F. W. (2016). Advanced driver-assistance systems: Challenges and opportunities ahead. Technical report, McKinsey & Company.

Wulfert, L., Kühnel, J., Krupp, L., Viga, J., Wiede, C., Gembaczka, P., and Grabmaier, A. (2024). Aifes: A next-generation edge ai framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(6):4519–4533.