Otimização de Sistemas IoMT Utilizando Sumarização de Vídeo em Tempo Real como Ferramenta de Inteligência em Dispositivos de Borda

Arthur Mendes Rocha Alves[1], **Fabiano Pereira Bhering**[1]

[1]Departamento de Computação
Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)
Campus Leopoldina – MG – Brasil

arthurmralves@gmail.com, fabiano@cefetmg.br

Abstract. *The surge in internet-connected devices has significantly increased data production, especially with the demand for real-time multimedia applications, like monitoring systems. This work proposes a video summarization technique utilizing edge computing to enhance Internet of Multimedia Things (IoMT) systems. Particularly in IoMT applications, the transmission of large amounts of multimedia data can strain communication infrastructure. The results show a reduction in network overload, maintaining a balance between computing and network resources without losing meaningful information.*

Resumo. *O avanço dos dispositivos conectados aumentou a geração de dados, especialmente com a demanda por aplicativos multimídia em tempo real em sistemas para Internet das Coisas Multimídia (IoMT). A transmissão de grandes quantidades de dados multimídia, como vídeos, pode sobrecarregar a infraestrutura e causar falhas de desempenho. Este trabalho propõe uma técnica de sumarização de vídeo utilizando computação nas bordas para aprimorar sistemas IoMT. Os resultados apresentam uma redução na sobrecarga da rede, mantendo o balanceamento entre os recursos computacionais e de rede.*

1. Introdução

A expansão das conexões de Internet das Coisas (IoT) impulsionou o surgimento da Internet das Coisas Multimídia (IoMT) [Alvi et al. 2015, Ericsson 2024]. Nesse cenário, o tráfego de vídeos representa um grande desafio, pois demanda alta largura de banda e rigorosas garantias de Qualidade de Serviço (QoS) [Bhering et al. 2022, Sabah and Selwal 2023]. O cumprimento desses requisitos é severamente dificultado nas redes sem fio da IoMT, que possuem capacidade limitada e estão frequentemente sujeitas a falhas de *link* e perda de pacotes [Ou et al. 2014].

Como grande parte das capturas de monitoramento contém fundos estáticos ou dados redundantes, o envio contínuo de vídeos brutos para processamento centralizado na nuvem sobrecarrega rapidamente a infraestrutura de comunicação [Hussain et al. 2019]. Para mitigar esse problema, torna-se essencial a adoção da computação nas bordas (*Edge Computing*) aliada a técnicas de sumarização de vídeo [Alaa et al. 2024, Zhang et al. 2016]. A sumarização recria uma representação breve e informativa do conteúdo, extraindo e retendo apenas os quadros visualmente relevantes [Hussain et al. 2021].

Neste contexto, o presente trabalho tem como objetivo propor e avaliar a utilização da sumarização de vídeo em tempo real como ferramenta de otimização na borda para sistemas IoMT. Ao processar e filtrar os dados na própria fonte geradora, a proposta visa reduzir a sobrecarga da rede, economizar largura de banda e viabilizar o funcionamento escalável e eficiente de aplicações multimídia.

2. Sumarização de Vídeo em Redes IoMT

Para embasar a proposta de otimização desenvolvida, esta seção revisa os princípios dos sistemas IoMT, computação na borda, assim como as abordagens de processamento e sumarização de vídeo.

2.1. Internet das Coisas Multimídia e Computação na Borda

A IoMT constitui uma evolução natural da IoT, integrando dados multimídia à rede de dispositivos [Nauman et al. 2020]. A arquitetura IoMT é caracterizada pela capacidade de oferecer suporte a dispositivos com severas restrições de custo, energia e recursos computacionais [Alvi et al. 2015].

Na base dessas redes, os nós sensores frequentemente operam com processadores baseados na arquitetura RISC, como a família ARM [Wang et al. 2019]. Essa arquitetura simplifica o conjunto de instruções, permitindo a redução significativa do consumo de energia e da emissão de calor [Lai et al. 2018]. Para contornar as limitações da infraestrutura de comunicação e não sobrecarregar as rotas, a estrutura da IoMT enfatiza a importância de deslocar o processamento dos dados multimídia para o mais próximo possível das bordas do sistema [Battisti et al. 2021].

2.2. Processamento e Sumarização de Vídeo

A sumarização de vídeo é a técnica utilizada para gerar um resumo informativo de um vídeo em um curto período de tempo [Meena et al. 2023]. Os resumos gerados podem ser classificados como estáticos, compostos pela seleção de quadros-chave isolados, ou dinâmicos, formados por pequenos fragmentos contínuos (*shots*) que garantem um resultado mais fluído [Bendraou et al. 2019].

As metodologias para seleção do conteúdo dividem-se em categorias principais: abstrativas, que geram novas representações sintetizadas do vídeo, e extrativas, que selecionam o conteúdo original com base em eventos, objetos ou áreas de interesse [Alaa et al. 2024, Dilawari and Khan 2019]. No contexto deste artigo, aplica-se a Sumarização de Visualização Única (*Single View Summarization* - SVS), que tem como objetivo resumir um arquivo de vídeo individualmente [Ji et al. 2020].

A execução dessa tarefa em tempo real, exigindo o processamento de capturas na taxa de 30 *frames* por segundo (FPS), impõe uma alta complexidade computacional [Shambharkar and Goel 2022]. Para viabilizá-la, o pré-processamento estrutural – como redimensionamentos e normalizações em escala de cinza – pode realizado de maneira otimizada por bibliotecas como o OpenCV [Uke et al. 2024] e YOLO (*You Only Look Once*)[Jocher and Qiu 2024], capazes de auxiliar na sumarização em tempo real, sendo ideal para aplicações em dispositivos de borda [Redmon et al. 2015].

2.3. Trabalhos Relacionados e Contribuições

A sumarização de vídeo em cenários de Internet das Coisas Multimídia (IoMT) tem evoluído de abordagens centralizadas para o processamento distribuído na borda, visando mitigar o gargalo de largura de banda e latência [Alaa et al. 2024, Meena et al. 2023]. A Tabela 1 apresenta uma análise comparativa entre propostas recentes da literatura e a solução desenvolvida neste trabalho.

Tabela 1. Comparação de técnicas de sumarização de vídeo para IoMT.

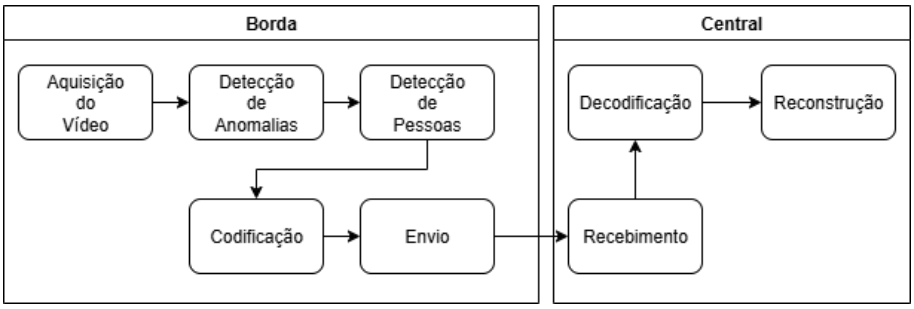
Trabalho	**Técnica de Sumarização**	**Processamento**	**Foco Principal**
[Ou et al. 2014]	Multi-view / Agrupamento	Sensores Visuais	Redundância
[Wang et al. 2019]	Redes Neurais (FANN)	Microcontrolador	Energia
[Uke et al. 2024]	YOLO + OpenCV	Borda (RPi)	Segurança
Proposta	**Diferenciação + YOLO**	**Borda (ARMv8)**	**Otimização IoMT**

A contribuição deste trabalho reside na proposição e validação de um modelo de processamento condicional assimétrico para dispositivos de borda em redes IoMT. Enquanto a literatura frequentemente foca ou na sumarização baseada em características visuais básicas ou no uso direto de redes neurais profundas, este trabalho demonstra que a combinação de um estágio de filtragem de baixa fidelidade (diferenciação de frames) com um estágio de validação de alta fidelidade (inferência via YOLOv11) cria uma simbiose técnica que resolve o dilema entre custo computacional e precisão em hardware restrito. A novidade não se limita à integração das técnicas, mas na demonstração de que a intermitência inteligente da rede neural, regida por um disparador de baixo custo, permite que nós sensores operem com uma economia de largura de banda de até 70% sem o *overhead* energético de uma execução contínua de IA. Portanto, é proposta uma arquitetura que viabiliza a inteligência artificial em tempo real em redes *mesh* de baixa taxa de transmissão, superando as limitações de latência de soluções centralizadas e a imprecisão de filtros de movimento puramente heurísticos.

3. Proposta e Desenvolvimento

A proposta deste trabalho é empregar um algoritmo de Sumarização de Visualização Única (SVS) dinâmico operando diretamente nos dispositivos de borda. O método baseia-se na extração de características-chave, priorizando movimento e a presença de objetos de interesse. O objetivo central é minimizar a carga na rede durante o envio de arquivos multimídia por conexões sem fio, condensando o conteúdo sem comprometer a essência das informações contextuais capturadas. O modelo foi desenvolvido com foco em dispositivos com baixo poder de processamento, mitigando a sobrecarga e otimizando a eficiência da transmissão e do armazenamento. O fluxo completo do sistema pode ser observado na figura 1

Figura 1. Funcionamento do sistema proposto.



3.1. Fluxo de Processamento na Borda

O processamento ocorre de forma autônoma nos nós sensores, seguindo um fluxo de seleção e filtragem antes de qualquer transmissão para a central. As etapas fundamentais para a seleção dos clipes de vídeo são descritas a seguir:

1. **Aquisição do vídeo:** A captura é realizada através de uma camera conectada via USB ao dispositivo ARM de borda, lendo cada *frame* individualmente dentro de um laço de repetição contínuo.
2. **Detecção de anomalias (Diferenciação de *frames*):** Para reduzir o custo computacional, o *frame* capturado é imediatamente convertido para escala de cinza. Calcula-se então a diferença absoluta entre as matrizes que representam o *frame* atual e o seu antecessor direto. Um limiar empírico (*movement_threshold*) determina se o quadro possui variação visual suficiente para justificar a continuidade do processamento.
3. **Detecção do objeto de interesse:** Utiliza-se a ferramenta de visão computacional YOLO para classificar os objetos presentes na imagem, com foco exclusivo na detecção de seres humanos. Se uma pessoa for detectada com uma confiança de pelo menos 50%, o *frame* e os quadros sucessores tornam-se elegíveis para envio.
4. **Envio dos *frames*:** Ao detectar o objeto com sucesso, aplica-se uma recarga (*cooldown*) nas inferências. Durante esse intervalo, os *frames* são enviados sequencialmente à central, que será a responsável por reconstruir o trecho (*shot*) recebido.

3.2. Codificação e Transmissão de Vídeo

Dadas as restrições da IoMT, a codificação dos *frames* é realizada inteiramente em memória, evitando operações de escrita e leitura no armazenamento secundário do dispositivo. Utiliza-se a função *imencode* da biblioteca OpenCV para compactar a matriz da imagem no formato JPEG. O resultado é um vetor de bytes armazenado de forma efêmera em um *buffer*, o que reduz substancialmente o tamanho do dado a ser transmitido e torna o envio via rede mais eficiente.

Para a transmissão, optou-se por comunicação ponto a ponto baseada no protocolo UDP, formando uma rede em malha. O dispositivo de borda envia inicialmente um datagrama de cabeçalho (24 bytes) contendo um inteiro como identificador único, *timestamps* de aquisição e envio, e o tamanho do quadro em bytes. Na sequência, o nó central recebe o *buffer* contendo a imagem codificada e remonta a sequência de *frames*.

3.3. Algoritmos de Sumarização e Recepção

Para viabilizar a sumarização em tempo real no hardware ARM embarcado, manipulações adicionais antecedem a inferência do YOLO. A conversão para escala de cinza reduz o peso do processamento em três vezes, operando sobre um único canal de cor ao invés de três (RGB). Adicionalmente, a resolução da imagem fornecida ao modelo é reduzida apenas no momento da inferência, exigindo menor poder computacional.

A lógica de controle do nó de borda está consolidada no Algoritmo 1, enquanto o fluxo de reconstrução na central encontra-se no Algoritmo 2.

Algorithm 1 Sumarização de Vídeo na Borda

while vídeo não terminou **do**
  Capturar próximo *frame*, incrementar ID e obter *timestamp*
  Converter *frame* para escala de cinza
  if *prev_frame* existe **then**
    Calcular diferença absoluta com o *prev_frame*
    if diferença $<$ *movement_threshold* **then**
      continue
    end if
  end if
  if *cooldown_timer* > 0 **then**
    Reduzir *cooldown_timer* e Enviar *frame* via UDP
    continue
  end if
  Reduzir a dimensão do vídeo para inferência
  Aplicar YOLO para detectar humanos
  if humano detectado **then**
    Aplicar *cooldown_timer* e Enviar *frame* via UDP
  end if
  Atualizar *prev_frame*
end while

Algorithm 2 Recepção de Vídeo na Central

while central ativa **do**
  Receber metadados e dados da imagem (*buffer*)
  Reconstruir *frame* (ignorar se inválido)
end while

4. Metodologia Experimental

Para garantir a reprodutibilidade dos resultados, os experimentos foram estruturados com base na calibração de parâmetros operacionais, visando o equilíbrio entre a sensibilidade de detecção e a eficiência computacional dos nós. O limiar de movimento (*movement_threshold*) foi definido empiricamente após testes preliminares para filtrar o ruído dos vídeos e variações de luminosidade que poderiam causar disparos. Complementarmente, estabeleceu-se um nível de confiança de 50% para a detecção de seres humanos via YOLOv11, valor que se mostrou eficaz para evitar falsos positivos sem comprometer a identificação de alvos em diferentes escalas e condições de oclusão parcial.

Cada experimento foi repetido múltiplas vezes sob as mesmas condições de rede para mitigar variações estocásticas inerentes ao protocolo UDP e à rede sem fio, sendo apresentadas as médias aritméticas das métricas monitoradas. A escolha das salas Conv_2B, Conv_4A e Conv_5A do dataset PIROPO foi estratégica por oferecerem cenários de monitoramento interno com desafios reais de iluminação, fundamentais para validar a robustez da sumarização.

Finalmente, a fidelidade semântica do conteúdo foi preservada através de um mecanismo de *cooldown* após a detecção positiva, o qual garante que frames sucessores ao evento de interesse sejam transmitidos sequencialmente à central, assegurando a continuidade temporal e a essência da informação visual mesmo com a redução da taxa de captura para 12 FPS.

4.1. Ambiente de Testes e Limites Computacionais

Os dispositivos de borda utilizados possuem arquitetura ARMv8-A de 64 bits com processador Cortex-A53 *quad-core*, operando entre 100MHz e 1512MHz, com 2GB de memória RAM e sistema operacional Armbian 5.10 (GNU/Linux). A conectividade Wi-Fi e USB permitiu a integração de uma *webcam* e a comunicação através de uma rede em malha sem fio previamente configurada com roteamento adaptativo.

Durante os experimentos, foram monitoradas quatro métricas de rede essenciais: latência, perda de pacotes, carga (*throughput*) e *jitter*. Inicialmente, realizaram-se testes para determinar os gargalos de processamento do dispositivo ARM, comparando a proposta com um *Baseline* de envio de vídeo normal, sem a técnica de sumarização.

5. Resultados

Os experimentos foram conduzidos para validar a eficiência da técnica proposta em um ambiente controlado e representativo das limitações reais de sistemas IoMT. As avaliações mensuraram o impacto da sumarização nas bordas tanto em relação à Qualidade de Serviço (QoS) da rede quanto ao consumo dos recursos de *hardware* dos sensores distribuídos.

5.1. Performance com Dataset (Vídeos Pré-Gravados)

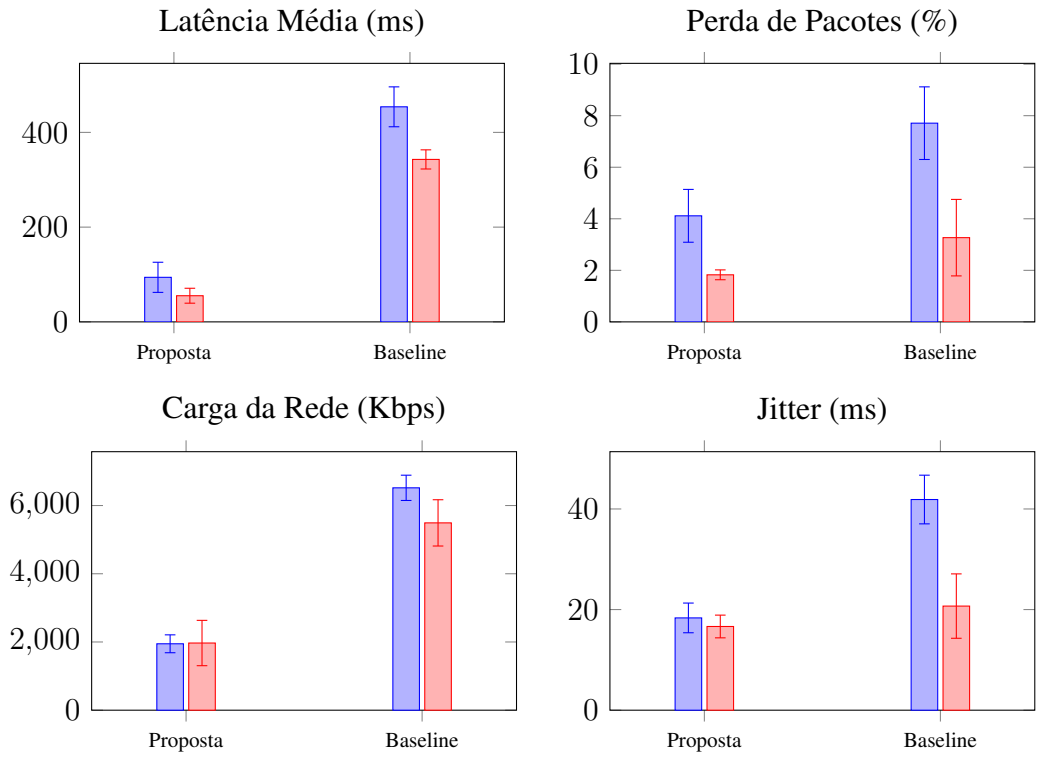
A primeira fase empírica avaliou a precisão e o impacto na rede utilizando a coleção de vídeos da base de dados PIROPO [del Blanco et al. 2021] (salas Conv_2B, Conv_4A e Conv_5A). Para manter a integridade visual, as imagens de resolução nativa (704x576 pixels) foram processadas e transmitidas. Os gráficos da Figura 2 indicam que a técnica de sumarização reduziu a carga na rede em até três vezes em comparação com o tráfego não sumarizado, oferecendo uma vasta economia de largura de banda.

A mitigação do congestionamento resultou em uma rede visivelmente mais confiável. Observou-se a redução da perda de pacotes, que caiu de valores próximos a 7,7% no *Baseline* para 4,1% com a sumarização. Da mesma forma, os níveis de *jitter* foram regularizados de forma drástica, com a variação média da latência (em Conv_2B) despencando de aproximadamente 41 ms para 18 ms.

5.2. Performance em Ambiente de Tempo Real

A segunda fase mensurou o modelo operando autonomamente via *webcam*. Para viabilizar a inferência ininterrupta no *hardware* embarcado, optou-se por reduzir a dimensionalidade dos *frames* de inferência para 128x128 pixels, estabelecendo a taxa de captura

Figura 2. Comparação de métricas de rede no conjunto de vídeos do Dataset PIROPO.



em 12 FPS. Os resultados dispostos na Figura 3 comprovam a aplicabilidade tática da arquitetura proposta.

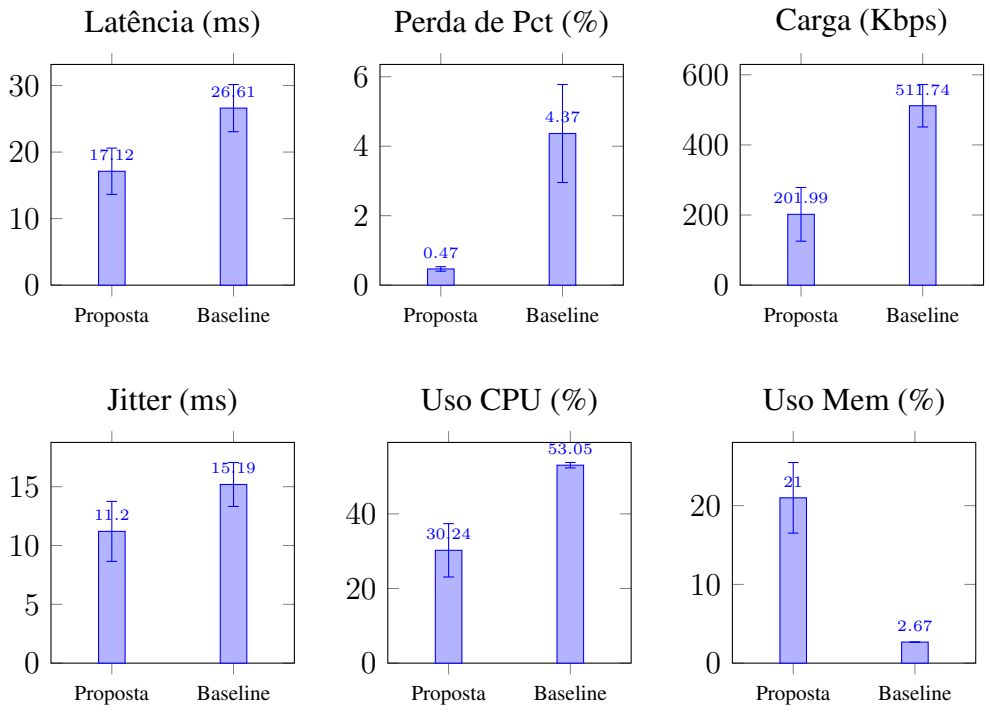
A latência média apresentou redução em relação ao padrão sem tratamento, acompanhada pela atenuação do *jitter* (de 15,18 ms para 11,20 ms), indicando que os pacotes não apenas chegam mais rápido, mas a consistência na estabilidade do fluxo inter-sistemas melhora consideravelmente. O indício mais promissor é a drástica redução na perda de pacotes, mitigada para 0,46%. Isso demonstra que a técnica contorna eficientemente a falta de mecanismos de garantia de entrega naturais do protocolo UDP, uma vez que a menor carga enviada à rede mitiga o transbordamento dos roteadores.

Na dimensão *hardware*, o *trade-off* mostrou-se favorável: a inteligência inserida na borda reduziu o uso médio da CPU de 53% para aproximadamente 30%, viabilizando a coexistência de outros serviços críticos no sensor, não obstante o previsível salto na utilização de RAM (de 2,6% para cerca de 21%) em decorrência da execução da rede neural YOLO.

5.3. Análise da Qualidade e Preservação da Informação

Na sumarização de vídeo em tempo real, é importante manter o equilíbrio entre a redução do volume de dados e a manutenção da utilidade da informação capturada. Neste trabalho, a preservação da fidelidade semântica é garantida pela lógica de operação do processamento em dois estágios. Ao adotar um limiar de confiança de 50% para a detecção de seres humanos, o sistema assegura que apenas eventos com alta probabilidade de relevância disparem a transmissão de dados. Adicionalmente, a implementação de um mecanismo

Figura 3. Desempenho da transmissão e processamento em tempo real (128x128 @ 12FPS).



de *cooldown* após a detecção positiva permite que frames sucessores ao gatilho inicial sejam enviados sequencialmente, o que preserva a continuidade temporal e o contexto do da presença humana, evitando que informações críticas sejam descartadas por flutuações momentâneas na detecção.

A proposta adota a estratégia de redução controlada de dimensionalidade. A conversão dos quadros para a resolução de 128x128 pixels é aplicada exclusivamente na fase de inferência do modelo YOLO, com o objetivo técnico de sustentar uma taxa de processamento estável de 12 FPS no hardware ARMv8 limitado. É fundamental destacar que essa compressão para análise não degrada necessariamente a reconstrução final na central, uma vez que o sistema é projetado para operar com foco na detecção de objetos de interesse em vez de fidelidade cinematográfica. Assim, a sumarização atua como um filtro semântico eficiente: embora a carga de dados enviada à rede seja drasticamente menor, a essência informativa necessária para aplicações de monitoramento e segurança é integralmente preservada, mitigando os efeitos negativos da sobrecarga de rede que, no cenário *baseline*, resultariam em perdas de pacotes superiores a 7%.

5.4. Discussão de Limitações

Embora a técnica reduza a carga de CPU de 53% para 30%, o uso de memória RAM apresentou um salto de 2,6% para 21%, representando um aumento significativo devido à carga do modelo YOLOv11. Esse aumento constitui um gargalo crítico para a escalabilidade horizontal do sistema, uma vez que dispositivos com hardware mais restrito poderiam enfrentar falhas por esgotamento de recursos ao tentar conciliar o sistema operacional com as exigências da rede neural. Além da restrição de memória, a abordagem baseada em diferenciação de frames é sensível a variações ambientais, como mudanças

bruscas de iluminação ou sombras, o que pode gerar disparos desnecessários do modelo de detecção e, consequentemente, desperdício de ciclos de CPU e energia. Por fim, a dependência do protocolo UDP para a transmissão dos trechos sumarizados, embora minimize a latência e o *overhead*, impõe o risco de perda de quadros ou entrega fora de ordem em ambientes com alta interferência, dificultando a reconstrução perfeita do conteúdo na central devido à ausência de mecanismos nativos de retransmissão.

6. Conclusão e Trabalhos Fututos

Este trabalho propôs a utilização da sumarização de vídeo na borda como estratégia para a otimização de sistemas IoMT. A abordagem combinou a detecção de anomalias por diferenciação de *frames* com o modelo YOLO para detecção de pessoas, garantindo que apenas quadros relevantes fossem transmitidos à central. Os experimentos, realizados em dispositivos de borda com arquitetura ARM, demonstraram reduções significativas no volume de dados transmitidos, na latência e na taxa de perda de pacotes. Esses ganhos evidenciam que a transferência da inteligência para as extremidades da rede contribui decisivamente para o funcionamento eficiente e escalável das aplicações multimídia.

Apesar do ganho expressivo na comunicação, a análise de limitações revelou que o custo da inteligência local reflete-se em um salto no consumo de memória RAM de 2,6% para cerca de 21%, o que impõe restrições de escalabilidade para dispositivos com recursos ainda mais severos. Além disso, embora o mecanismo de *cooldown* tenha garantido a preservação da fidelidade semântica dos eventos de interesse, a dependência do protocolo UDP em redes *mesh* permanece sensível a cenários de alta interferência.

Como trabalhos futuros, planeja-se investigar técnicas de quantização e poda (*pruning*) do modelo YOLOv11, visando reduzir a pegada de memória RAM e viabilizar a execução em dispositivos de borda com hardware ainda mais restrito[cite: 149]. Pretende-se também realizar uma avaliação experimental comparativa abrangente, incluindo outros algoritmos de sumarização da literatura e diferentes modelos de redes neurais, como CNNs de baixo custo e arquiteturas baseadas em *Transformers* otimizadas para a borda. Além disso, vislumbra-se a implementação de métricas objetivas de qualidade visual, como PSNR e SSIM, para quantificar com maior precisão o grau de degradação dos frames reconstruídos na central. Por fim, projeta-se a expansão do sistema para suportar a detecção de múltiplas classes de objetos em ambientes externos variados e a integração de um serviço de armazenamento em nuvem escalável para a consulta histórica dos vídeos sumarizados.

Referências

Alaa, T., Mongy, A., Bakr, A., Diab, M., and Gomaa, W. (2024). Video summarization techniques: A comprehensive review.

Alvi, S. A., Afzal, B., Shah, G. A., Atzori, L., and Mahmood, W. (2015). Internet of multimedia things: Vision and challenges. *Ad Hoc Networks*, 33:87–111.

Battisti, A. L. É., Muchaluat-Saade, D. C., and Delicato, F. C. (2021). Enabling internet of media things with edge-based virtual multimedia sensors. *IEEE Access*, 9:59255–59269.

Bendraou, Y., Essannouni, F., and Salam, A. (2019). From local to global key-frame extraction based on important scenes using svd of centrist features. *Multimedia Tools and Applications*, 78(2):1441–1456.

Bhering, F., Passos, D., Ochi, L. S., Obraczka, K., and Albuquerque, C. (2022). Wireless multipath video transmission: when iot video applications meet networking—a survey. *Multimedia Systems*, 28(3):831–850.

del Blanco, C. R., Carballeira, P., Jaureguizar, F., and García, N. (2021). Robust people indoor localization with omnidirectional cameras using a grid of Spatial-Aware classifiers. *Signal Process. Image Commun.*, 93(116135):116135.

Dilawari, A. and Khan, M. U. G. (2019). ASoVS: Abstractive summarization of video sequences. *IEEE Access*, 7:29253–29263.

Ericsson (2024). Ericsson mobility report. Technical report, Ericsson. Accessed: YYYY-MM-DD.

Hussain, T., Muhammad, K., Del Ser, J., Baik, S. W., and de Albuquerque, V. H. C. (2019). Intelligent embedded vision for summarization of multiview videos in iiot. *IEEE Transactions on Industrial Informatics*, 16(4):2592–2602.

Hussain, T., Muhammad, K., Ding, W., Lloret, J., Baik, S. W., and de Albuquerque, V. H. C. (2021). A comprehensive survey of multi-view video summarization. *Pattern Recognit.*, 109(107567):107567.

Ji, Z., Xiong, K., Pang, Y., and Li, X. (2020). Video summarization with attention-based encoder–decoder networks. *IEEE Trans. Circuits Syst. Video Technol.*, 30(6):1709–1717.

Jocher, G. and Qiu, J. (2024). Ultralytics yolo11.

Lai, L., Suda, N., and Chandra, V. (2018). CMSIS-NN: efficient neural network kernels for arm cortex-m cpus. *CoRR*, abs/1801.06601.

Meena, P., Kumar, H., and Kumar Yadav, S. (2023). A review on video summarization techniques. *Eng. Appl. Artif. Intell.*, 118(105667):105667.

Nauman, A., Qadri, Y. A., Amjad, M., Zikria, Y. B., Afzal, M. K., and Kim, S. W. (2020). Multimedia internet of things: A comprehensive survey. *Ieee Access*, 8:8202–8250.

Ou, S.-H., Lee, C.-H., Somayazulu, V. S., Chen, Y.-K., and Chien, S.-Y. (2014). On-line multi-view video summarization for wireless video sensor network. *IEEE Journal of Selected Topics in Signal Processing*, 9(1):165–179.

Redmon, J., Divvala, S. K., Girshick, R. B., and Farhadi, A. (2015). You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640.

Sabah, A. and Selwal, A. (2023). Data-driven enabled approaches for criteria-based video summarization: a comprehensive survey, taxonomy, and future directions. *Multimedia Tools and Applications*, 82:32635–32709.

Shambharkar, P. G. and Goel, R. (2022). From video summarization to real time video summarization in smart cities and beyond: A survey. *Front. Big Data*, 5:1106776.

Uke, S., Junghare, P., Kenjale, S., Korade, S., and Kothwade, A. (2024). Comprehensive real-time intrusion detection system using iot, computer vision (opencv), and machine

learning (yolo) algorithms. In *2024 4th International Conference on Ubiquitous Computing and Intelligent Information Systems (ICUIS)*, pages 1680–1689.

Wang, X., Magno, M., Cavigelli, L., and Benini, L. (2019). Fann-on-mcu: An open-source toolkit for energy-efficient neural network inference at the edge of the internet of things. *CoRR*, abs/1911.03314.

Zhang, K., Chao, W.-L., Sha, F., and Grauman, K. (2016). Video summarization with long short-term memory. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*, pages 766–782. Springer.