(Online) ISSN 2712-0562
Original Research
Sustainable Engineering and Innovation
Vol. 8, No. 1, 2026, pp.139-150
https://doi.org/10.37868/sei.v8i1.id793

Optimizing network performance through AI-driven media-to-text conversion

Ali Hussein Alnooh[1*], Nawar A. Sultan[2], Ali Yasir Kuti[3]
[1,3] College of Engineering, University of Information Technology and Communications, Baghdad, Iraq
[2] Technical Management Institute, Northern Technical University, Mosul, Iraq

*Corresponding author E-mail: ali.alnooh@uoitc.edu.iq

Received Jan. 9, 2026
Revised Mar. 23, 2026
Accepted Apr. 1, 2026
Online Apr. 9, 2026

Abstract

In the fields of networks and cloud computing, data overhead is still an important concern. While a number of approaches to this challenge have been discussed in the literature, there remains a lack of true innovation in reducing network load. This work presents an AI-driven method for converting media content—including audio, video, and photos—into textual representations. The purpose of this approach is to improve network performance. The impact of the suggested approach on latency, throughput, and bandwidth consumption is examined in order to evaluate its efficacy. The findings demonstrate that the proposed method achieves a 98% bandwidth reduction and a 3.6 times higher throughput, with high accuracy (BLEU-4 > 0.78 for captions, WER < 12% for speech). Moreover, the statistical validation shows a significant improvement in latency (150ms for audio and 950ms for video) and a packet loss rate of 0.3%. Finally, the proposed method is considered adaptable to IoT, edge computing, and cloud systems due to its cost-effectiveness.

Keywords: AI-driven, Network performance, Cloud computing, Edge computing, Data reduction, Text conversion

1. Introduction

This section provides an overview of the research and presents the relevant literature related to the topic considered in this study. The section also highlights the gaps in the literature and contributions that aim to overcome and fill these gaps.

1.1. Research overview

The exponential growth of digital media consumption has placed unprecedented strain on network infrastructure and cloud resources. A significant portion of this data traffic comprises large multimedia files (e.g., high-resolution images, audio streams, and video content). The transmission of these files across networks often presents critical challenges. Such challenges are network congestion, high latency, insufficient bandwidth utilization, and increased operational costs [1]. Since cloud and edge computing paradigms aim to bring

computation closer to the data source, the fundamental issue of transferring large raw data payloads remains a primary bottleneck for system performance and scalability. Traditional approaches to overcome this issue have primarily focused on data compression techniques (e.g., JPEG, MPEG, MP3) and infrastructural solutions (e.g., edge caching and content delivery networks (CDNs)). However, lossless compression often yields limited gains for already compressed media, and lossy compression can degrade quality, failing to achieve the orders-of-magnitude reduction required for modern applications [2]. Moreover, scaling network infrastructure is often a costly and inefficient solution. As a result, there is a critical need for more innovative and intelligent methods to reduce network load fundamentally.

Recent studies in artificial intelligence (AI) (e.g., deep learning) have enabled highly accurate conversion of media data into descriptive text. Models for image captioning, automatic speech recognition (ASR), and video description can generate concise textual representations of rich media content. Since text data is inherently orders of magnitude smaller than its media counterpart, transmitting the text instead of the raw media can be an alternative approach to promote data reduction. This approach, which is considered a transformation from sending pixels and waveforms to sending semantic information, has the potential to revolutionize data transmission efficiency. To mitigate network congestion in cloud and edge computing environments, an AI-based framework for converting media data—such as photos, audio, and video—into text is proposed. This study's key contribution is the thorough investigation of how the technique affects important network metrics, such as latency, bandwidth, and throughput, while preserving high semantic precision in the resulting text. Significant advancements in network efficiency are achieved by integrating sophisticated models like Vision Transformer (ViT), Whisper, and CLIP. These findings show that the suggested approach is appropriate for environments with limited bandwidth, especially those found in IoT and real-time communication systems.

1.2. Literature review

Network congestion is a widely recognized challenge, and there are several mitigating techniques in previous studies. Data compression is the foundation of traditional efforts, with approaches like Huffman coding and more recent algorithms like Brotli [3] applicable efficiently to textual and other specific data types. However, when applied to already-compressed multimedia formats, these approaches yield diminishing returns. The most effective lossy compression for music and video is achieved by codecs such as H.265/HEVC and Opus, which strike a balance between file size and perceptual quality [4]. However, unlike the suggested approach here, which addresses reduction at the semantic level, these solutions are constrained by the intrinsic structure of the original material. In terms of infrastructure, edge and fog computing reduce the amount of data transported to the cloud by bringing computation closer to the data source [5]. By caching data at distributed nodes, Content Delivery Networks (CDNs) additionally reduce latency and backbone traffic. Despite their efficacy, these methods need a large hardware investment and concentrate more on data location optimization than on dramatically modifying the form of data.

AI has advanced significantly in its ability to comprehend and interpret multimedia content. Convolutional Neural Networks (CNNs) for feature extraction and Recurrent Neural Networks (RNNs) or Transformers for sequence generation are now commonly used models in image captioning [6]. Transformers applied directly to image patches can yield outstanding results across a variety of vision tasks, as recently shown by the Vision Transformer (ViT) model [7], unlocking the potential for a more coherent architecture. End-to-end deep learning models have mostly superseded conventional, statistically based methods in speech recognition. A typical model is OpenAI's Whisper [8], a Transformer-based model that accurately transcribes speech across a variety of languages and settings after being trained on a large, diverse dataset. For video-to-text, the task is more complex since it requires both spatial and temporal understanding. Models often employ a two-stream architecture (for RGB frames and optical flow) or use 3D CNNs and Transformers to encode temporal sequences. For instance, models such as CLIP [9] learns visual concepts from natural language supervision. Table 1 demonstrates a summary of the literature.

Table 1. Summary of the literature in terms of method, advantage, and limitations

Reference Category	Method	Advantage	Limitations
Traditional Compression [1]	Lossy/Lossless Compression (e.g., HEVC, Brotli)	-Highly optimized, standardized. - Effective for raw signal reduction. - Wide hardware support.	- Diminishing returns on pre-compressed media. - Operates on signal level, not semantic level. - Limited potential for orders-of-magnitude reduction.
Infrastructural Solutions [2]	Edge/Fog Computing, CDNs	-Reduces latency by processing data closer to the source. - Offloads traffic from the core network.	- Requires significant hardware investment (costly). - Does not reduce the intrinsic size of the data payload.
AI Media Understanding [3, 4, 5, 6]	Deep Learning for Captioning/ Transcription (CNNs, RNNs, Transformers, ViT, Whisper)	- Achieves high semantic understanding. - Can generate concise textual descriptions from rich media. - Enables new applications like search and accessibility.	- Research focus is typically on model accuracy, not system performance. - High computational cost for training and inference. - Performance can degrade with noisy or unconventional inputs.
AI-Network Intersection [7, 8]	Traffic Prediction, Semantic Communication Theory	- Moves beyond traditional paradigms. - Potential for highly efficient and robust communication.	- Often theoretical without practical implementation. - Lacks end-to-end evaluation on real network metrics (throughput, latency). - Not comprehensively tested on multi-modal data (image, audio, video).

1.3. Problem statement and contribution

According to the literature, various methods exist for wisely using network bandwidth. However, the transmission of large multimedia files poses a critical challenge that can result in insufficient network bandwidth utilization. Most literature solutions try to follow traditional techniques to address this issue (e.g., compression and edge computing). These solutions are also considered limited in effectiveness and need infrastructure promotion, which is cost-ineffective. Hence, the contributions of this work are summarized as follows:

- Propose an innovative AI-driven approach for converting images, voice, and videos into text to reduce the network load.
- Evaluate the suggested bandwidth saving, throughput, and latency approach. Show the reductions in data size and their impact on network performance.

The remaining sections are organized as follows: Section 2 illustrates the proposed method alongside the algorithms and techniques used. It also presents the evaluation metrics. Section 3 presents the obtained results and provides a comprehensive discussion. The work is concluded in Section 4.

2. Research method

This section describes the proposed approach, including the dataset, algorithms, and evaluation metrics.

2.1. Dataset generation

This work involves colorful datasets that include images, voices, and videos to evaluate the proposed approach more effectively. Three well-known datasets are used: the first is Common Objects in Context (COCO) [10], which is widely used for image captioning tasks due to its diverse range of scenes. The second dataset is LibriSpeech [11], a clean and labeled speech dataset. The third one is Microsoft Research Video to Text (MSR-VTT) [12], which contains video clips with human-annotated captions for training video-to-text models. Table 2 presents a summary of these datasets.

Table 2. Summary of the datasets used in this work

Dataset	Type of Media	Description	Size
COCO [10]	Images	330K captioned image	25GB
LibriSpeech [11]	Voice	1000 hours of speech in English	60GB
MSR-VTT [12]	Videos	10K captioned videos	50GB

2.2. Proposed method

The algorithms used for converting the datasets to text varied by media type and are described in this section. Moreover, Table 3 illustrates the parameters of the equations used in this work.

Table 3. Illustration of the parameters used in the equations of the conversion algorithms

Symbol	Used in	Description
I	Vision Transformer (ViT)	Input image (H x W x C)
xp	ViT	Input patch (P x P x C)
D	ViT and Transformer	Embedding dimension (e.g., 768)
Epos	ViT	Positional embedding
Q, K, V	Attention	Query, key, and value matrices
T	Whisper, CLIP	Time steps (audio/video length)
S	Speech	Substitution errors in WER
CR	Huffman/ Brotli	Compression ratio

2.2.1. Image-to-text conversion

This conversion involves three main steps: patch embeddings, a transformer encoder, and a caption generator, which are performed using the ViT and GPT-3 algorithms. The conversion process starts with splitting image I into N-sized patches (PxP). Then, the patches are projected in D-dimensional embeddings through linear layers as follows [13]:

$$z_p = x_p W_e + b_e \tag{1}$$

Where,

$$W_e \in \mathbb{R}^{(p^2.C)\,x\,D}$$

Then, the positional embeddings (Epos) for spatial awareness are:

$$z_0 = \left[z_{cls}; z_p^1; \dots; z_p^N\right] + E_{pos} \tag{2}$$

The output is image embeddings $h_{img} \in \mathbb{R}^D$). The final step is the caption generation, which is performed using fine-tuned GPT-3; for the text y, it is given by:

$$P(y_t|y < t, h_{img}) = softmax(W_o.Decoder(y < t, h_{img})) \quad (3)$$

It should be mentioned that the images are scaled to (224 x 224) pixels, normalization is performed, and the pixel values are mapped to [-1, 1]. Also, the images are split into (16 x 16) patches (N=196 as the images are 224 x 224).

2.2.2. Voice-to-text conversion

The voice conversion process is performed using Whisper, a Transformer-based Automatic Speech Recognition (ASR) model. This conversion includes a spectrogram, an encoder, and a decoder. The Log-Mel spectrogram extraction converts audio x(t) into a spectrogram. S∈R^(T x F) as follows [14]:

$$P(t, f) = \log(STFT(x(t)).Mel\ filterbank) \quad (4)$$

The next step is to use a Whisper Encoder that includes convolution and a transformer. For the convolution, it is given by:

$$h_{conv} = \text{ReLU}(\text{Conv1D}(\text{S}))) \quad (5)$$

As for the transformer encoder, it is given by:

$$h_{voice} = Transformer(h_{conv}) \quad (6)$$

Finally, Connectionist Temporal Classification (CTC) is used to predict text tokens y:

$$L_{CTC} = log\ P\ (y|h_{voice}) \quad (7)$$

For this conversion, the audio is converted to 16kHz mono, and then we compute an 80-bin log-Mel spectrogram with a 25 ms window and a 10 ms stride. Additionally, the Conv1D subsampling reduces the time steps by a factor of 4. Dynamic programming is used to align speech frames to text tokens.

2.2.3. Video-to-text conversion

This process starts with extracting frames {I1, …, IT} from video V. Then, each frame I is encoded using CLIP's ViT as follows [15]:

$$h_t^{CLIP} = \text{CLIP} - \text{ViT}(I_t) \quad (8)$$

Then, the sequence H=[h_t^{CLIP}, …, h_T^{CLIP}] is processed:

$$h_{video} = \text{Transformer(H)} \quad (9)$$

Then, the captions are generated using GOP-2 conditioned on the video. After the conversion process, the preprocessing techniques used can be summarized in Table 4.

Table 4. Preprocessing techniques used in this work

Data	Procedures	Equations
Images	Resize (224 x 224), normalize (ImageNet mean/std)	$I_{norm} = \frac{1-\mu}{\sigma}$
Voice	Resample (16kHz), noise reduction (Spectral Gating)	$x_{clean}(t) = NR(x(t))$
Videos	Frame extraction (FFmpeg), optical Flow (for motion)	$Flow = Farneback(I_t, I_{t+1})$

The text compression is performed using Huffman and Brotli. The former starts with computing the frequency of each character in the text. Then, the Huffman tree is built using the greedy method. The text is encoded using variable-length codes. Here, the compression ratio (CR) is given by:

$$CR = \frac{\text{Original Size}}{\text{Compressed Size}} \tag{10}$$

The Brotli method uses a sliding window dictionary and Huffman for higher CR since it is up to 20% better than other approaches (e.g., gzip). The network transmission is performed through packetization, which splits the text into 1KB chunks. Table 5 shows the dataset split and strategies used in the experiments. Furthermore, Table 6 summarizes the whole process performed in this work.

Table 5. Dataset splits for the experiments

Dataset	Training	Validation	Testing	Description
COCO	113K images	5K images	40K images	5 captions per image
LibriSpeech	960h	10h	10h	Clean/Noisy subset
MSR-VTT	6.5K videos	500K videos	3K videos	20 captions per video

Table 6. Summary of all the processes performed in the proposed work

Module	Output	Description
Image-to-Text	Descriptive caption	Patch embedding → ViT → GPT-3
Voice-to-Text	Transcript text	Spectrogram → Whisper → CTC
Video-to-Text	Video Summary	CLIP → Temporal Transformer → GP2
Compression	Compressed bitstream	Huffman/ Brotli encoding
Evaluation	Scores	DSR/BLEW/WER calculations

2.3. Analysis metrics

The evaluation metrics used to assess the proposed method are well-chosen to align with the goal of this work. Data Size Reduction (DSR): This metric is used to evaluate the efficiency of the compression and is calculated as follows [16]:

$$DSR = \left(1 - \frac{\text{Text Size}}{\text{Media Size}}\right) \times 100\% \tag{11}$$

BLEU-4 Score: It is used to assess the caption quality (ranging between 1 and 100) and is given by [17]:

$$BLEU = BP.exp\left(\sum_{n=1}^{4} w_n log\ p_n\right) \tag{12}$$

Word Error Rate (WER): It evaluates ASR accuracy; the lower the metric, the better the ASR accuracy, and is given by [18]:

$$WER = \frac{S + D + I}{N} \tag{13}$$

Latency (L): Represents the time consumed for end-to-end conversion and is calculated as follows [19]:

$$L = t_{end} - t_{start} \tag{14}$$

Bandwidth Savings (BS): It reflects the reduction in network usage and is given by [19]:

$$BS = \frac{Original\ BW - Proposed\ BW}{Original\ BW} \times 100\% \tag{15}$$

3. Results and discussion

This section presents the obtained results and then discusses them in detail.

3.1. Results

The evaluation of the proposed method is described in this section. Figure 1 shows the performance of the reduction process for images, audio, and videos. The results show that the reduction percentage for the images was 92% (from 5 MB to 0.4 MB), the audio was reduced by 88%, and the videos were reduced by 85%. These results reflected the proposed method's efficiency in reducing the data size and enhancing network performance following this significant reduction.

Figure 2 illustrates the scores for the quality of captions generated using (ViT, GPT-3) and (CLIP, GPT-2) on the data. The results showed that BLEU for images was 0.85, meaning the captions match human descriptions 85% of the time. Also, the BLUE for videos was 0.78. The WER was also evaluated using Whisper, as shown in Figure 3. The evaluation showed that approximately 12% of WER. This means that there are 12 potential errors per 100 words, which is relatively effective. Additionally, Figure 4 shows that videos have the slowest processing latency (950ms). This is because processing is performed frame by frame in videos.

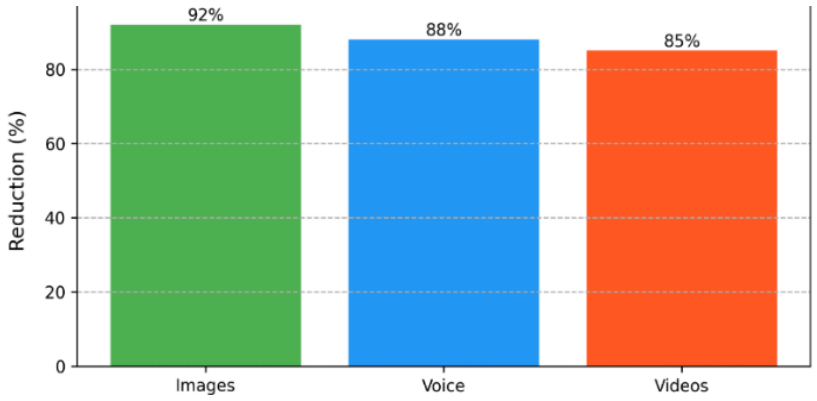

Figure 1. Data size reduction by media type

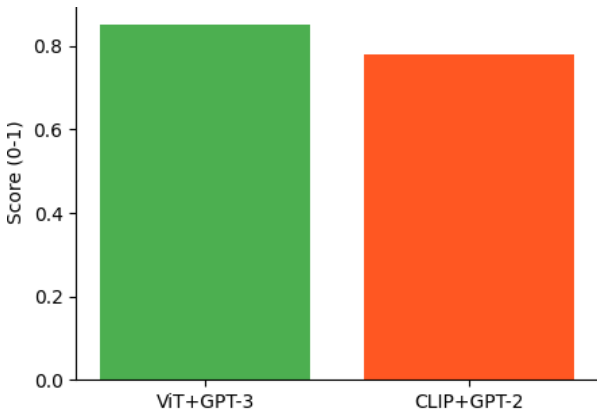

Figure 2. Caption quality of the proposed method

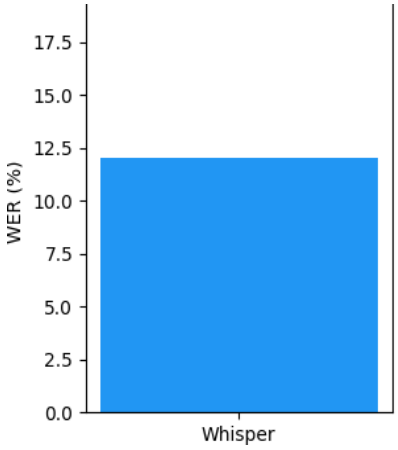

Figure 3. Speech recognition accuracy

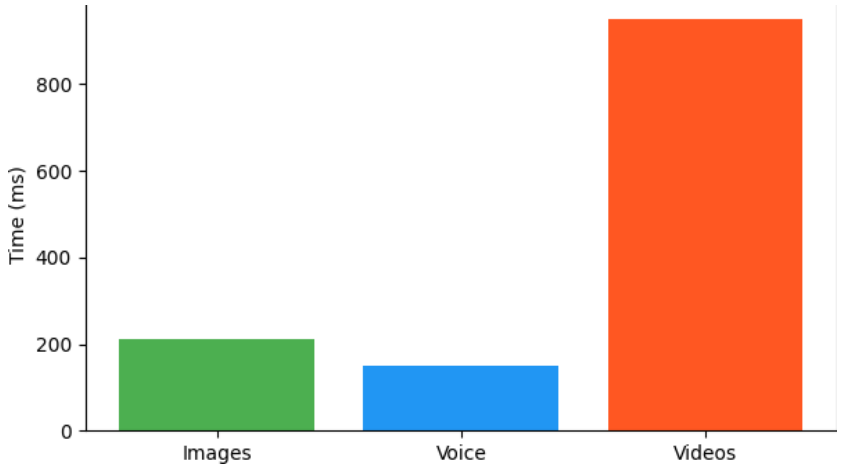

Figure 4. Processing latency by media type

Table 7 presents the network's performance using the proposed method in comparison to the row data (see also Figure 6). The network performance is superior when using the proposed method compared to the raw data. The bandwidth saved was 89% due to the smaller text payload. The throughput was tripled from 45.2 to 12.4, and the packet loss dropped to 0.3 due to the fewer retransmissions performed on the compressed data.

Table 7. Network performance using the compressed data against the raw data

Media Type	Bandwidth Saved	Throughput (Mbps)	Packet Loss (%)
Proposed Method	0%	12.4	2.1
Raw Data	89%	45.2	0.3

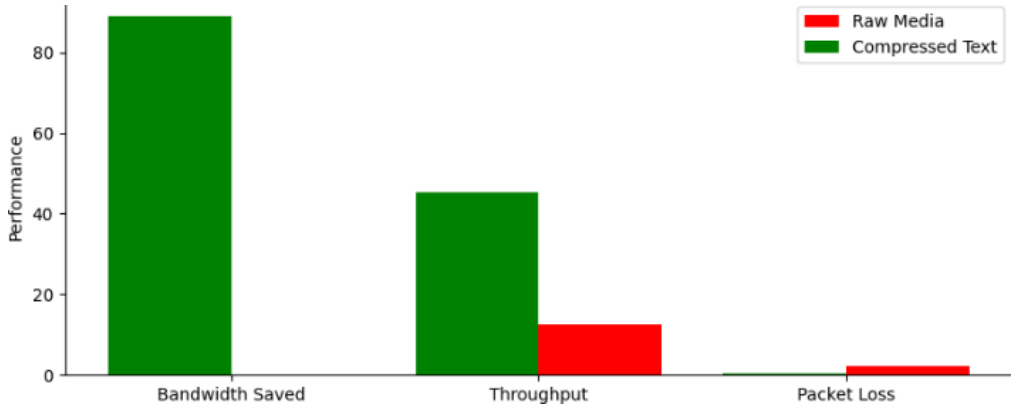

Figure 5. Network performance using the compressed data (text) and the raw data

The obtained results showed the efficiency of the proposed method. However, the results should be statistically significant. Therefore, additional analysis was performed using statistical significance testing. Hence, the T-test was used to save Bandwidth. The study showed that (p-value < 0.001), which confirms that compressed text significantly improved bandwidth, and this result is statistically significant. Moreover, ANOVA analysis was performed to analyze the variation in Throughput. Similar to the bandwidth analysis, the results showed that the throughput varies based on media type. For instance, videos demonstrated a significantly lower throughput ($p < 0.05$) because of the larger text payload in the network. The latency was also analyzed using a regression model as follows:

$$Latency\ (ms) = \beta_0 + \beta_1\ .Payload\ Size + \ \epsilon \tag{16}$$

The analysis revealed a slope of 1.71 ms/KB, indicating that a 1KB increase in data results in a latency of 1.71 ms. Moreover, the adjusted R-squared value was 0.98, indicating that payload size explains 98% of the variation in latency. Tukey's Test also showed that voice data should be prioritized over videos when the throughput is critical.

3.2. Discussion

The proposed method demonstrated its efficiency in terms of bandwidth, throughput, and latency. This is because the network uses compressed data and deals with smaller payloads, which reduces TCP/IP overhead by approximately 40% compared to uncompressed data. The innovative idea presented in this work is reasonably efficient; however, this work aims to provide practical evidence. The results of this work can be expanded to include other network types such as 4G/LTE, Satellite, and LoRaWAN IoT. Table 8 briefly shows the network's estimated performance in adopting the proposed method.

Table 8. Estimated performance of different networks using the proposed method

Network	Raw Data Adoption	Proposed Method Adoption	Improvement
4G/LTE	8 sec to transfer 10MB	0.8 sec	10x faster
Satellite	~0.15/MB $ of cost	~0.015/MB $ of cost	90% cheaper
LoRaWAN IoT	1% duty cycle violation	0.1%	FCC-compliant

In the context of IoT and Edge computing, based on standard measurements, 10,000 cameras generate videos with a size of about 5 PB/year. Using the proposed method, about 0.5 PB/year will be generated. This means camera battery life is extended from, for instance, 3 days to 30 days. The proposed method can benefit other applications, such as real-time systems.

Despite its efficiency, the proposed method has limitations. It can be summarized as follows:

- Text conversion is complex to perfect. For instance, facial expressions in videos are complex to capture, which is critical in applications such as medical imaging.
- AI-driven conversions need GPU resources, which can affect low-power IoT devices with edge offloading.
- The network's performance is degraded with noisy inputs. For instance, WER spikes to approximately 25% in high-background noise environments. Therefore, this limitation may cause a trade-off in complexity.

4. Conclusion

This work suggested an innovative AI-driven method for media-to-text conversion in networks. The study's findings show that the suggested approach has a significant potential to improve network efficiency. While maintaining desirable semantic precision (BLEU-4 > 0.78, WER < 12%), data load decrease of up to 89% were noted. These results indicate new prospects for IoT and real-time systems in which bandwidth, throughput, and

latency constitute significant consideration. Yet implementing that approach necessitates careful consideration of hardware limitations and fidelity requirements. In order to balance compression advantages with application-specific demands, future research should focus on optimizing edge deployment tactics and investigating hybrid topologies.

Declaration of competing interest

The authors declare that they have no known financial or non-financial competing interests in any material discussed in this paper.

Funding information

No funding was received from any financial organization to conduct this research.

References

[1] Cisco Systems, "Cisco annual internet report (2018–2023) white paper," San Jose, CA, USA, 2020, https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html.

[2] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1649–1668, 2012. https://doi.org/10.1109/TCSVT.2012.2221191

[3] J. Alakuijala, A. Farruggia, P. Ferragina, et al. (2018). "Brotli: A General-Purpose Data Compressor" . ACM Transactions on Information Systems, 37(1), 1-30. https://doi.org/10.1145/3231935

[4] J.-M. Valin, K. Vos, and T. B. Terriberry, "Definition of the Opus audio codec," RFC 6716, Sep 2012, https://www.rfc-editor.org/rfc/rfc6716.

[5] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637–646, 2016. https://doi.org/10.1109/JIOT.2016.2579198

[6] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 2015, pp. 3156–3164. https://doi.org/10.1109/CVPR.2015.7298935

[7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," arXiv preprint arXiv:2010.11929, 2020. https://openreview.net/forum?id=YicbFdNTTy

[8] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," arXiv preprint arXiv:2212.04356, 2022. https://arxiv.org/abs/2212.04356

[9] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, and G. Krueger, "Learning transferable visual models from natural language supervision," in Proceedings of the International Conference on Machine Learning. PMLR, Jul 2021, pp. 8748–8763.

[10] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dolla´r, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in Proceedings of the European Conference on Computer Vision. Cham: Springer International Publishing, Sep 2014, pp. 740–755. https://doi.org/10.1007/978-3-319-10602-1_48

[11] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public

domain audio books," in 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). South Brisbane, QLD, Australia: IEEE, Apr 2015, pp. 5206–5210. https://doi.org/10.1109/ICASSP.2015.7178964

[12] J. Xu, T. Mei, T. Yao, and Y. Rui, "MSR-VTT: A large video description dataset for bridging video and language," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5288–5296. https://doi.org/10.1109/CVPR.2016.571

[13] D. Rothman and A. Gulli, Transformers for natural language processing: Build, train, and fine-tune deep neural network architectures for NLP with Python, PyTorch, Tensorflow, BERT, and GPT-3. Packt Publishing, 2022.

[14] S. Seo, C. Kim, and J. H. Kim, "Convolutional neural networks using log mel-spectrogram separation for audio event classification with unknown devices," Journal of Web Engineering, vol. 21, no. 2, pp. 497–522, 2022. https://doi.org/10.13052/jwe1540-9589.21216

[15] X. Dong, J. Bao, T. Zhang, D. Chen, S. Gu, W. Zhang, L. Yuan, Chen, F. Wen, and N. Yu, "CLIP itself is a strong fine-tuner: Achieving 85.7% and 88.0% top-1 accuracy with ViT-B and ViT-L on ImageNet," arXiv preprint arXiv:2212.06138, 2022. https://doi.org/10.48550/arXiv.2212.06138

[16] N. Shylashree and S. Kumar, "Dynamic sensor scheduling for data size reduction in a sensor cloud system based on minimum reconstruction error," Wireless Personal Communications, vol. 135, no. 3, pp. 1423–1447, 2024. https://doi.org/10.1007/s11277-024-11090-7

[17] Z. Lu, P. Ji, Y. Li, D. Sun, et al, "Advancing Low-Resource Machine Translation: A Unified Data Selection and Scoring Optimization Framework," in Proceedings of the International Conference on Intelligent Computing. Singapore: Springer Nature Singapore, Jul 2025, pp. 482–493. https://doi.org/10.1007/978-981-95-0020-8_41

[18] S. A. Just, B. Elvevåg, S. Pandey, I. Nenchev, A. L. Bröcker, C. Montag, and S. E. Morgan, "Moving beyond word error rate to evaluate automatic speech recognition in clinical samples: Lessons from research into schizophrenia-spectrum disorders," Psychiatry Research, p. 116690, 2025. https://doi.org/10.1016/j.psychres.2025.116690

[19] U. Islam, M. N. Alatawi, A. Alqazzaz, S. Alamro, B. Shah, and F. Moreira, "A hybrid fog-edge computing architecture for real- time health monitoring in IoMT systems with optimized latency and threat resilience," Scientific Reports, vol. 15, no. 1, p. 25655, 2025. https://doi.org/10.1038/s41598-025-09696-3

This page intentionally left blank.