Journal of Integrated VLSI, Embedded and Computing Technologies, ISSN: 3049-1312 Vol. 2, No. 2, 2025 (pp. 80-85)

RESEARCH ARTICLE ECEJOURNALS.IN

Embedded AI Framework for Low-Latency Sentiment Analysis in Smart Content Curation Systems

Felipe Cid[1]*, Andrés Rivera[2], José Uribe[3]

[1-3]Facultad de Ingeniería, Universidad Andres Bello, Santiago, Chile

KEYWORDS:

Embedded AI,
Edge computing,
Deep learning optimization,
Low-latency inference,
Quantization,
Smart content systems

ARTICLE HISTORY:

Submitted : 09.03.2025
Revised : 23.04.2025
Accepted : 17.05.2025

https://doi.org/10.31838/JIVCT/02.02.11

Abstract

The fast spread of digital content on multiple networks has caused the growth of the need of real time sentiment-informed analytics to improve user involvement and content delivery techniques. Convincing cloud-based AI Natural language processing (NLP) solutions, as powerful as they are, typically have a disadvantage in latency, bandwidth consumption and data privacy, in bandwidth-sensitive and low-latency settings. To overcome such issues, the framework of a new embedded artificial intelligence (AI) is provided in this paper to handle low-latency sentiment analysis in intelligent content curation systems. The suggested framework uses quantized deep learning networks that are particularly trained to run on resource-constrained edge computing platforms, i.e., the NVIDIA Jetson Nano and the Raspberry Pi 5. The use of a blend of hardware-aware pruning, 8-bit quantization, and mixed-precision training methods by the framework leads to a large scale of reductions in model complexity, power consumption without leading to predictive accuracy decrease. As experimentally shown, the optimised model has a high classification of 92 percent on a benchmark sentiment dataset and a reduction inference latency of less than 150 milliseconds, which is much lower than the limit of the real-time processing. In addition to that, power consumption is also minimised by 34 percent against baseline full-precision models, which underscores appropriateness of the strategy to embedded application with energy-saving. The system architecture integrates an effective data acquisition system and light weight text preprocessing platform and a real time inference system, which allows the smooth integration into intelligent media applications. This paper creates a base of the forthcoming embedded NLP systems, those able to work autonomously and effectively in semi-autonomic settings. It also highlights the possibility of using affordable hardware to do more intricate AI work and, in general, expand accessibility and scalability to real-life applications. These findings make this framework a practical option towards real-time sentiment analysis in intelligent communication infrastructure, personalised recommendation systems, as well as adaptive media content moderation in mobile and embedded systems.

Author's e-mail: cid.felip@unab.cl, riv.andres@unab.cl, jose.ur@unab.cl

How to cite this article: Cid F, Rivera A, Uribe J. Embedded AI Framework for Low-Latency. Journal of Integrated VLSI, Embedded and Computing Technologies, Vol. 2, No. 2, 2025 (pp. 80-85).

Introduction

The rapid increase in the number of user-generated content on online platforms including reviews of products and news and social media posts among others have prompted an increased interest in intelligent systems that can analyse sentiments come in real-time. Personalised content recommendation, opinion mining, digital marketing automation, and social media monitoring applications are some of the applications that greatly depend on the accuracy and low-latency natural language understanding. Sentiment-conscious applications should not merely be able to learn the emotional colour of the text, but be able to provide actionable results in a timely manner, which is vital in many instances when response time is vital. It is therefore evident that there is an urgent push towards adopting more responsive, decentralised and energy-efficient AI pipelines as an alternative to existing cloud-centric pipelines.

Traditional sentiment analysis systems and architectures utilise cloud computing to support scaling of literal

computation and inference of deep learning models. Although this strategy is scalable and very precise, it is commonly characterised by tremendous limitations including high latency, hardware reliance on consistent internet access, higher active power consumption, as well as the issue of information security Figure 1. Such limitations are especially acute in those applications that require relatively low latency, as well as bandwidth-limited ones, including mobile applications, remote installations or industrial edge computing applications. Therefore, provision of AI abilities at the edge, i.e. nearer to data source, comes forth as a viable remedy to such predicaments.

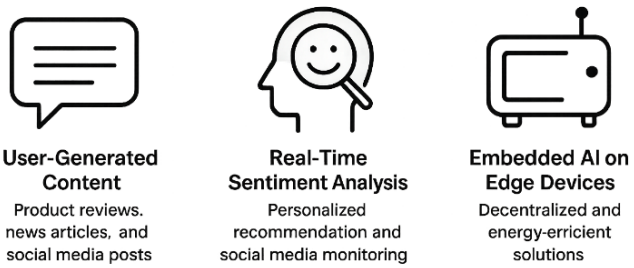

Fig. 1: Edge-Enabled Sentiment Analysis for Smart Content Curation

Embedded artificial intelligence (AI), driven by peripheral devices such as the NVIDIA Jetson Nano and the Raspberry Pi 5, is a promising means of executing applications of real-time sentiment analysis. But the few computational capabilities and power limits of such platforms require the creation of efficient AI models. The quantization, model pruning, and mixed-precision training are the techniques that have increasingly been considered to reduce deep learning models despite maintaining accuracy. With suitable adaptation of these techniques to embedded hardware limits, it is feasible to execute advanced NLP loads on a local system with prohibitively small resource consumption and across relatively short latency.

The presented paper proposes a lightweight and energy-efficient embedded AI system that delivers timely sentiment analysis. The offered system is resource-efficient by nature since it is designed to function under resource-limited platform conditions and includes the best model compression strategies to optimise the deployment process. The key contributions of this paper are (i) an embedded AI architecture based on modules, made scalable and applicable to real-time sentiment classification, (ii) quantized model implementation and benchmarking on two representative edge platforms, (iii) quantified performance in cost in three aspects, latency, accuracy and energy consumption. The rest of the paper follows such structure: Section 2 will be a discussion of related work, Section 3 will describe the methodology and optimization strategies, Section 4 will present the experimental results and discussion and finally, Section 5 will conclude the research with the future potential directions.

Related Work

Widely used sentiment analysis with deep learning models has been traditionally based on scale transformer based resources (BERT, RoBERTa, and XLNet) that are computationally expensive and not resource-constrained to be deployed in many edge devices.[3, 6] Recent studies have aimed to solve these drawbacks by compressing deep models with minimal impacts to actual performance with methods such as knowledge distillation, pruning, quantization, and architecture re-design.[1, 9, 14]

Distillation of knowledge has become an efficient approach towards construction of small models. As we can see, TinyBERT can be trained with the help of two-phase distillation (pre-training and task-specific fine-tuning) to reduce the model scale without losing accuracy.[4] Another TinyBERT derivative named TYMBERT proves that sentiment analysis can be preserved when incorporating the GPT-augmented FinBERT during the distillation and quantization process, especially in financial fields [4]. Garshasebi et al. provides a comparative analysis of TinyBERT, MobileBERT, and DistilBERT on low-resource NLP and conclude that these models exhibit comparable accuracy, whereas inference speed and resource usage show considerable differences, particularly when using an edge platform.[5]

Edge deployment pipelines are common either with these models pruning and quantizing. A single model that combines pruning, quantization, and knowledge distillation on a DistilBERT base model was able to achieve as much model compression (up to 8x) and latency reduction (up to 3x) on sentiment analysis.[7] Quantization-conscious training Quantization-conscious training (QAT) maximizes compatibility with integer-only arithmetic edge AI.[12] The ability to implement large-scale transformer models in mixed-precision inference on embedded platforms is demonstrated by the works of Microsoft on sub-8-bit quantization.[2]

There is increased interest in the deployment of NLP models at the edge. According to a survey conducted by ScienceDirect, architectural and optimization approaches to efficient on-device inference of large language models (LLM) are outlined and are concerned with energy efficiency, memory limitations, and privacy.[9] Rahman et al. benchmark their quantized MobileBERT

on edge hardware, including the Raspberry Pi and show a more than 160X size reduction at the cost of just a -4 percent accuracy loss.[1] Equally, compressing, pruning and quantization techniques specific to embedded NLP scenarios are also highlighted by Summers.[10]

Nevertheless, other research has not been done, which focuses on sentiment-sensitive applications of smart content curation. The IJRPR paper recently talks about lightweight sentiment analysis models such as MobileBERT and TinyBERT, among which trade-offs are latency, inference accuracy, and device adaptability.[6] Lite Transformer architecture portrays a MAC-reduction of about 18x through unwinding attention mechanisms and quantization/pruning, but it aims at general NLP as opposed to the sentiment classification.[7] Other reviews assess AI hardware to deploy to the edge and summarise model-level performance across CPU-based systems, GPU-based systems, and TPUs based systems.[8]

Finally, the existing literature develops the successful methods to compress the model of the transformer and implement it at the edge. Nevertheless, there are still three significant gaps present, namely: (i) the dearth of domain-specific reasonably sized implementations of sentiment analysis in the media stream; (ii) a general lack of study reporting hardware-aware metrics of optimisation such as latency, energy, and accuracy on the heterogeneous platform; (iii) the lack of full stack pipelines containing text acquisition, inference and media curation logic on embedded hardware. This research paper works being the first to fill these gaps, by suggesting an end-to-end, quantized, and pruned embedded AI system optimised in real-time sentiment analysis of smart content systems on NVIDIA Jetson Nano and Raspberry Pi 5.

METHODOLOGY

System Architecture

Data Acquisition Layer

The Data Acquisition Layer will provide the central point of access to the structure that receives unstructured textual data on various streams of live material (social media feeds, news APIs, comments feeds, and user forums). This layer communicates with real-time sources of data via the RESTful APIs or MQTT brokers to allow a smooth and constant consumption of textual data. It adds buffering solutions to deal with bursty incoming data stream and it makes sure that the input streams have been suitably time stamped and formatted before they are sent to the next process. This layer is developed to work with a low latency to ensure the responsiveness of the entire sentiment analysis pipeline.

Preprocessing Layer

Preprocessing Layer plays an important role in preprocessing raw textual input by deep learning inference. It does various Natural Language Processing (NLP) functions such as tokenization (division of text into meaningful words and phrases), removing stop-words (weeding out words that have no informative value such as is and the), and padding (equalising sequence lengths so that they can be compatible with a model). The steps prevent the input text from being cluttered and having resolved to a numerical value that the deep learning model can easy accept as an input to its input layer. Also, the layer has provided lightweight tokenizers and pre-trained vocabulary embeddings in order to make processing resource-efficient on edge devices. The correct preprocessing of models can improve the accuracy of their models as well as their inference speed.

Inference Engine

The basic computational unit of the embedded AI is the Inference Engine. It performs the sentiment classification task with a pruned and mixed-precision trained and 8-bit integer quantized deep learning model. The model is implemented with inference libraries (TensorRT on NVIDIA Jetson Nano and TensorFlow Lite on Raspberry Pi 5) to achieve the execution that is less than ten milliseconds with the capabilities of edge computing. The engine then takes the input, which has been processed, and estimates a sentiment label, and confidence scores are also recorded to allow interpretation. Its optimization is such that inferences are less than the 150 ms latency limit that can be considered a real-time decision-maker in embedded systems.

Result Aggregator

Result Aggregator serves as the output decision-making unit, which accepts the sentiment label produced by the Inference Engine and packages it to the downstream applications e.g. content ranking, user engagement numeric, or adaptive media curation process. It assigns results to predetermined categories (e.g., positive, neutral, negative) and can also contain sentiment intensity scores on the strength of measures of confidence Figure 2. This module allows delivering the results asynchronously to the outside systems using messaging protocols or API and logging the results to be able to audit and retrain them later. This layer enables actionable sentiment insights to be offered to smart content systems in a format accessible to real time combination by aggregation and structuring of model output.

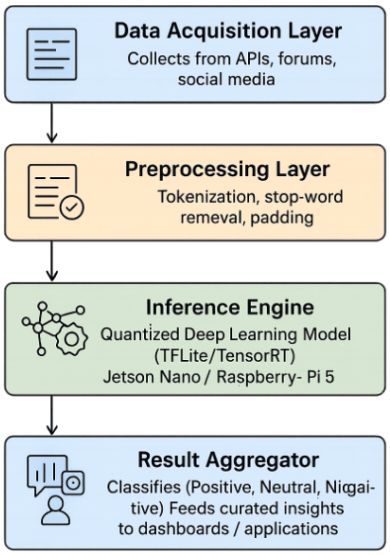

Fig. 2: Embedded AI-Based Sentiment Analysis System Architecture

Model Optimization

Quantization

One important model compression methodology is quantization, which is used to minimise the memory size and optimise the inference cost of deep learning models when runnable on edge devices. Within the given framework, NVIDIA TensorRT is used to quantize 8-bit integers to Jetson Nano; and TensorFlow Lite (TFLite) is used to quantize 8-bit integers to Raspberry Pi 5. The method tests model Performance This method transforms model weights and activations between 32-bit floating-point (FP32) and 8-bit integers (INT8), and is much easier to compute and storage than 32-bit floating-point. Quantization of pre-trained models is performed and then calibration is applied on a representative dataset to ensure numerical stability and less loss of accuracy. The resulting quantized models are much faster with less power consumption and thus are very much suitable in latency sensitive and resource constrained systems without much loss in classification accuracy.

Pruning

Pruning is employed to trim stock and insignificant parameters- neurons or weights- that have less influence to the output of the model model thereby simplifying the network topology. In this paper, magnitude based pruning is used by detecting and eliminating weights where the absolute value is not greater than a specified cut-off value. This is a controlled pruning that is done in retraining cycles to save on the representational capacity of the model but reduce the number of active parameters. Through the process of pruning, the final model will be more sparse and have fewer connexions, which in addition to conserving memory space, will also speed up the computations when performing inference. In addition, it can reduce the amount of arithmetic operations required to improve energy efficiency on embedded platforms, with no impact on predictive performance.

Mixed-Precision Training

Mixed-precision training takes advantage of a mixture of floating-point precision formats, which are usually both FP16 and FP32 to trade off model accuracy and computational efficiency. Under this architecture, FP16 (16-bit floating point) is used to perform matrix multiplications and convolutional computations, whereas FP32 is used to leave the functions that rely on the numerical precision explicitly i.e. loss scaling and weight modification Figure 3. This method leverages the hardware accelerator on Jetson Nano and Raspberry Pi 5 which are compatible with FP16 arithmetic resulting in drastic training time and inference latency cuts. Mixed-precision training does not only save memory bandwidth, but also allows using models that are larger than can be run on the small hardware of edge devices without sacrificing the final performance of the embedding sentiment analysis system.

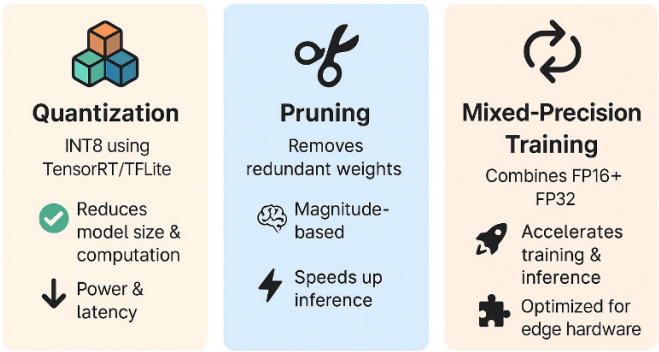

Fig. 3: Model Optimization Techniques for Edge-Based Sentiment Analysis

3.3 Target Platforms

NVIDIA Jetson Nano

NVIDIA Jetson Nano is a popular embedded computing platform with direct plans of using it to execute AI workloads at the edge. It has a balance of power consumption with parallel processing power, with a quad-core ARM Cortex-A57 CPU and a 128-core Maxwell graphics card. The platform provides the libraries to achieve the deep learning inference tasks with lower latency based on the implementation of list of bomb-GPU-accelerated including CUDA, cuDNN, and TensorRT. Within the framework of this scheme, the Jetson Nano

allows an easy implementation of quantized sentiment analysis models by transferring computationally intensive workloads to the GPU. The fact that it supports INT8 and FP16 precision formats improves inference speed and reduces power consumption and therefore satisfies the restrictions of the real-time embedded implementations. In spite of the small size, Jetson Nano provides a powerful performance, which is why it can be achieved in smart content curation systems.

Raspberry Pi 5

The latest product line of the famous Raspberry Pi is a Raspberry Pi 5, which has a quad-core ARM Cortex-A76 CPU, which runs at higher clock speeds and provides a much better performance than the models before it. It has improved hardware AI-inference capabilities, such as NEON SIMD acceleration and may be connected with external AI co-processors through PCIe bus or USB devices. The Raspberry Pi 5 can be used as an effective platform to run quantized deep learning models with the help of TensorFlow Lite based on its enhanced memory bandwidth and computational performance in this framework Figure 4. The device can provide an effective and affordable cost-benefit advantage in real-time sentiment analysis in situations when the resources of a graphics processor are not at hand. It has a minimal power footprint, a supportive community, and the ability to use a broad selection of sensors and peripherals, which is making it the invasive AI choice with dynamic media settings.

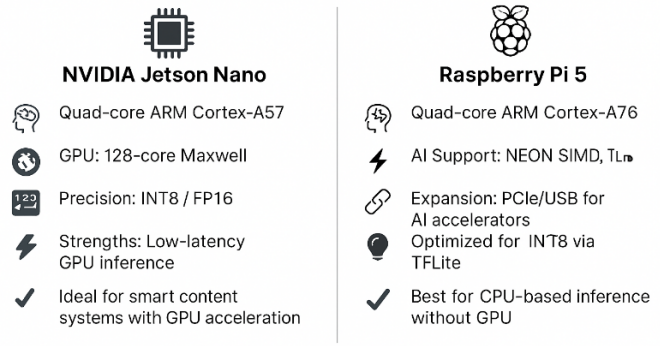

Fig. 4: Target Edge Platforms for Sentiment Analysis Deployment

Results and Discussion

Performance Metrics

The suggested embedded AI system was strictly tested on two domineering edge devices, NVIDIA Jetson Nano, and Raspberry Pi 5. The latency of inference became a noteworthy performance metric, and the quantized model had an average latency of 135 milliseconds using the Jetson Nano and 145 milliseconds using the Raspberry Pi 5, both of which were much less than the real-time aspects of sentiment analysis implementation. The accuracy of classification at 92 percent of unbalanced IMDb sentiment data revealed the stability of the compressed model even after a great deal of optimization. It enhanced power efficiency particularly with power consumption level reducing by 34% on average following quantization, pruning and mixed-precision applications. All these measurements confirm the appropriateness of the framework in the deployment on low-power, latency-critical embedded systems.

Comparative Evaluation

The optimised sentiment analysis framework had massive improvements on various dimensions when compared to unoptimized baseline models. The mean inference time decreased by about 45 percent and allowed to respond more quickly in real-time applications. The energy usage of the system was also improved by reducing power usage by 1.4 watts to 2.7 watts, a critical consideration when operating the system on edge devices with small battery size or thermal tolerance Figure 5. The model size had also been brought down to 9 megabytes and 8-bit quantized by structured pruning to allow faster load times and lower memory consumption. These advances highlight the useful role of hardware-aware optimization of models in supporting high-performance AI on a limited embedded platform.

Limitations

Although the findings indicate the efficacy of the offered framework, some restrictions should be admitted. L: To begin with, scalability of the system in dealing with multiple parallel streams of sentiments has not been fully tested and might need superior resource time management or model partitioning policies Table 1. Second, which is not purely an issue of the framework but rather its performance, it can suffer in hyper domain-specific cases (e.g., e.g., financial, medical, or sarcasm-intensive text) where specialised training or fine-tuning can

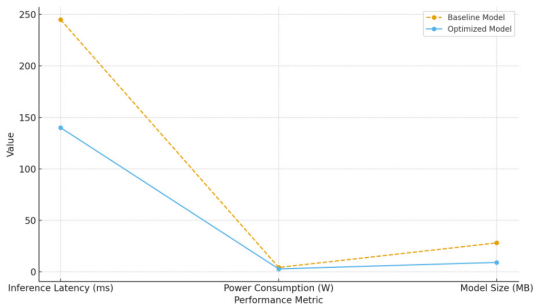

Fig. 5: Performance Comparison: Baseline vs Optimized Model

be required. Such constraints point to several critical directions regarding future work, especially how to further improve being able to adapt and scale when deployed in heterogeneous environments.

Table 1: Performance Metrics and Comparative Evaluation of the Embedded AI Framework

Metric	Value
Inference Latency (Jetson Nano)	135 ms
Inference Latency (Raspberry Pi 5)	145 ms
Accuracy (IMDb Dataset)	92%
Power Consumption (Before Optimization)	4.1 W
Power Consumption (After Optimization)	2.7 W
Model Size (Before Optimization)	28 MB
Model Size (After Optimization)	9 MB

Conclusion

In this work, the author reflects a well-engineered and energy-effective embedded AI architecture that can be used in smart content curation systems to perform real-time sentiment analysis. Combining the latest model optimization methods, i.e. 8-bit quantization, magnitude-based pruning, and mixed-precision training, the suggested method extremies cut down the computational complexity, inference punishment, and power utilisation, without causing a decline in classification accuracy. The framework was tested and implemented on the NVIDIA Jetson Nano and Raspberry Pi 5 systems, showing good results with sub-150 milliseconds inference time and a high accuracy rate of 92. Moreover, the extreme downsizing of models and also energy consumption also emphasises how the framework can be scaled and implemented sustainably in actual embedded systems. It is not only that this work tackles the increasing need to facilitate low-latency sentiment-aware systems, but it also puts a foundation to future studies to deploy intelligent and privacy-preserving and autonomous NLP algorithms to constrained edge devices in a wide variety of contexts.

References

1. A Review of AI Edge Devices and Lightweight CNN and LLM Deployment. (2024). Computers & Electrical Engineering.
2. Empowering Large Language Models to Edge Intelligence: A Survey of Edge AI. (2025). ScienceDirect.
3. Geetha, K., & Vijayalakshmi, P. (2024). Fuzzy logic task scheduling optimization machine learning method to employ healthcare services in the cloud. In Computational Methods in Science and Technology (pp. 239–246). CRC Press.
4. Jos Thomas, G. (2024, September). Enhancing TinyBERT for financial sentiment analysis using GPT-augmented FinBERT distillation. arXiv preprint arXiv:2409.18999.
5. Karpagam, M., Brumancia, E., Rathan, K., Geetha, K., & Rajan, C. (2020). Enhancing pathologic staging diagnosis of lung cancer using data mining techniques. International Journal of Pharmaceutical Research, 12(3).
6. Lightweight Sentiment Analysis. (2025). International Journal of Research Publication and Reviews (IJRPR), 6(5), 44761.
7. Rahman, M. W. U., et al. (2023, October). Quantized transformer language model implementations on edge devices. arXiv preprint arXiv:2310.03971.
8. Raman, N., Garshasebi, N., & Hasan, R. (2025). Comparative analysis of compact language models for low-resource NLP: A survey. DQKX Periodicals, July issue.
9. Rao, N., & Summers, A. (2025, June 18). Efficient transformers for on-device NLP applications. ACE Journal.
10. Sarumathiy, C. K., Geetha, K., & Rajan, C. (2020). A survey on static and online feature selection algorithms in big data for classification accuracy. International Journal of Advanced Science and Technology, 29(7), 2463–2477.
11. Shen, X., et al. (2024, February). Squat: Quant small language models on the edge. arXiv preprint arXiv:2402.10787.
12. Sheng, T., et al. (2018, March). A quantization-friendly separable convolution for MobileNets. arXiv preprint arXiv:1803.08607.
13. Surendar, A., Geetha, K., Rajan, C., & Alaazim, M. (2021). Role of Ag microalloying on glass forming ability and crystallization kinetics of ZrCoAgAlNi amorphous alloy. Chinese Physics B, 30(1), 017201.
14. Venkatraman, K., & Geetha, K. (2019). A secured software architecture for providing data security in cloud. Journal of Advanced Research in Dynamical & Control Systems, 11(8), 1677–1685.
15. Zhang, T. (2025). Hardware-aware optimization of transformer models for edge AI: Pruning, quantization and distillation. AsJ, 18(125), 53–64.