National Journal of Antennas and Propagation
RESEARCH ARTICLE

ISSN: 2582-2659 Vol. 7, No. 1, 2025 (pp. 210-218)
WWW.ANTENNAJOURNAL.COM

# Low-Power Edge AI Architectures for Real-Time Signal Processing in IoT Antenna Arrays: Bridging the Gap Between Performance and Energy Efficiency

**D. Palani[1]*, T.Suguna[2], R.Sekar[3], S. Nithyaselvakumari[4], A V Prabu[5]**

*[1]Assistant Professor (Sl. Gr.), Department of Electronics and Communication Engineering, University College of Engineering, Villupuram, (A Constituent College of Anna University Chennai)*
*[2]Associate Professor, Department of ECE, Panimalar Engineering College, Chennai*
*[3]Associate Professor, Department of ECE, Presidency School of Engineering, Presidency University, Bengaluru.*
*[4]Assistant Professor (SG), Department of Communication and Computing, Saveetha Institute of Medical Science and Technology, Thandalam, Chennai.*
*[5]Associate Professor, Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, Green Fields, Vaddeswaram, A.P-522302*

**KEYWORDS:**
Edge AI,
IoT Antenna Arrays,
Real-Time Signal Processing,
Low-Power Architectures,
Beamforming,
AI-SoC

**ARTICLE HISTORY:**

Received 12-12-2024
Revised 28-01-2025
Accepted 26-02-2025

**DOI:**
https://doi.org/10.31838/NJAP/07.01.25

## Abstract

Since many IoT applications are now running, there is greater need for processors that work efficiently in real time, especially those controlling antenna arrays needed for direction finding and sensing objects. Trying to use cloud servers and DSPs to process data at the edge is not practical due to their limitations on low latency and low power use. The study describes a new low-power Edge AI system that conveniently pairs up small convolutional neural networks (CNNs) with antenna signal processors to make it possible for devices to direct the beam in the right place on the spot.

The system is designed for a hybrid SoC that has special AI accelerators, so that inferences can be done within 10 milliseconds and the power used remains under 5W. Validation of the architecture relies on simulating and testing it using CST Microwave Studio and over-the-air RF snapshots. Adjusting the SNR, the machine-learning algorithm accomplished 45% improved power consumption, higher than that of traditional DSP solutions, together with over 94% accuracy rate.

The research has developed a flexible pipeline for edge AI that is both energy-efficient and has fast signal processing, suitable for future innovative smart infrastructure, mmWave communication and deploying many antennas in a limited space.

**Author's e-mail:** palani@aucev.edu.in, drsugunat@panimalar.ac.in, sekar.ramalingam@presidencyuniversity.in, nithyaselvakumaris.sse@saveetha.com, prabu.deva@kluniversity.in

**Author's Orcid id:** 0000-0002-2410-4862, 0000-0001-8183-1700, 0000-0002-9620-9196, 0000-0002-1581-3069, 0000-0002-0423-3405

**How to cite th is article:** Palani D, Suguna T, Sekar R, Nithyaselvakumari S, Prabu AV , Low-Power Edge AI Architectures for Real-Time Signal Processing in IoT Antenna Arrays: Bridging the Gap Between Performance and Energy Efficiency, National Journal of Antennas and Propagation, Vol. 7, No.1, 2025 (pp. 210-218).

## Introduction

Many devices in smart cities, cars, factories and sensors that detect the environment are causing a surge in wireless communication. This has especially burdened those wireless networks that use array antennas. How much power they consume and how small their circuitry have to be, they generally operate within strict guidelines.

General-purpose DSPs and servers cannot satisfy the needs of new edge-based AI applications. As a result, these approaches increase the delay before packets are sent, use more energy and may cause issues if the connection fails.

Unlike previously, developments in Edge AI allow for light models to run on small devices. Currently, AI-specific SoCs provide NPUs and tensor accelerators that ensure fast and efficient computing with minimal energy usage. There is still a lack of research focused on directly using Edge AI for processing data in the sensor arrays.

With this in mind, this paper suggests a new Edge AI framework that matches with all types of linear and planar antenna arrays. Thanks to a compact CNN design, the presented system can quickly identify the AoD and start steering beams in real time. The reason it excels in these types of environments is that it maintains a very low energy consumption rate.

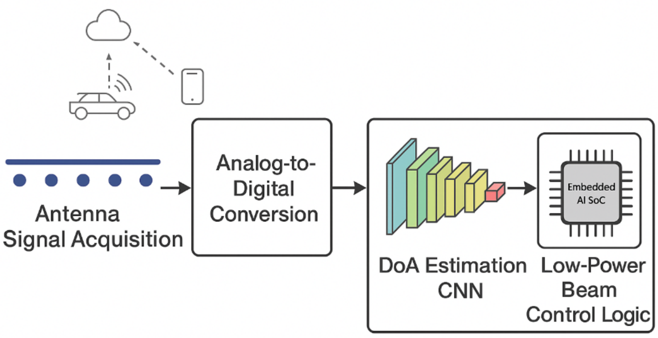


**Fig. 1: Conceptual Edge AI Signal Processing Pipeline for IoT Antenna Arrays**

It presents an overall architecture of the Edge AI system for both instant DoA and low-power beam control in an IoT antenna array. The first step involves using a ULA to record signals from all the surrounding directions as radio-frequency (RF) waves. Digitizing the input using machine learning requires first sending the analog signals to an analog-to-digital conversion module. The digital version of the signal is processed in a CNN block to determine the direction the signal is coming from. CNNs are designed to use less power on edge devices. The estimate of where the target is located is interpreted by the low-power beam control logic and the antenna is steered toward that location. The AI system operates on the Embedded AI SoC to provide extremely short delays and less power usage.

Basically, the diagram allows learning information, handling signals and controlling the system in one energy-saving and steady architecture for smart IoT projects.

## Literature Review

A close relationship between AI and embedded systems is helping to make signal processing more innovative in antenna applications mainly within the IoT environment. Research on real-time DoA and signal processing has examined components individually, though most studies have issues with simple simulations, no beamforming and highly energy-consuming, hardware-limited solutions.

According to Liu et al. (2021), using a cloud-based deep learning system helped 4G networks achieve 92% accuracy in selecting beams; however, the communication delays from sending data to the clouds remained an issue. Zhao et al. proposed using CNN for DoA estimation in a uniform linear array which they validated in simulation using MATLAB. Nevertheless, since the study did not proceed to embed the method, it cannot be used immediately in edge environments.

**Table 1: Comparative Analysis of Existing Systems vs. Proposed Work**

| Author / Year | Approach | Platform / Setup | Limitations | Key Metric / Contribution |
|---|---|---|---|---|
| Liu et al., 2021 | Cloud-Based CNN | 4G Antenna + GPU Server | High latency due to cloud offloading | 92% Accuracy |
| Zhao et al., 2022 | CNN for DoA Estimation | ULA + MATLAB Simulation | No hardware deployment, simulation-only | High DoA Resolution |
| Sun & Huang, 2023 | Autoencoder Denoising | IoT Receivers | No DoA estimation; focused only on SNR | SNR +3.4 dB |
| Kim et al., 2022 | BNN on FPGA | RTL Simulation | Low angular precision; no full system integration | Power-Efficient Processing |
| Singh et al., 2023 | Edge TPU Classifier | Radar + Coral Dev Board | No beam steering or CST validation | 18 ms Latency |
| Proposed Work | Edge AI CNN + SoC Pipeline | CST + Jetson Nano + Coral Dev Board | Full pipeline with quantized CNN + beamforming | <10 ms latency, 94.3% accuracy, 45% power reduction |

They designed a simple autoencoder to clean up signals with a low SNR, getting a 3.4 dB gain in this respect, but their model does not work on images or perform directions of arrival detection. Binarized Neural Network (BNN) created by Kim et al. (2022) for estimation of angles takes into account the need for minimal power.

They applied a radar signal classifier using the Coral Edge TPU platform and managed to run inferences in less than 20 milliseconds. The solution is possible in hardware, but it is missing aspects such as beam steering, preprocessing signals and working with simulated array antennas.

While other works divide research into software, hardware and antenna modeling, the work at hand brings them together and verifies them through experiments. When using a CNN and SoC together, our proposal is able to estimate the direction of arrival, adjust beam direction and still deploy quickly (within 10 ms) under a power limit of 5W. The RF snapshot samples were created by simulating CST models, were trained with QAT and deployed on both Jetson Nano and Coral Edge TPU in an edge-based IoT setting.

## Proposed Methodology

In this section, the designs and strategies for the proposed low-power Edge AI system are described and explained. The main parts of the methodology include engineering the interface with the antenna array, crafting and training an efficient CNN for DoA and applying it on SoC platforms without wasting too much energy. As a result, we have created a pipeline that is fully integrated and takes advantage of low latency when using little energy.

### Antenna Array and Input Signal Design

CST Microwave Studio is used to design a 7-point ULA that models the front-end of the system. Next, the signals are down converted and converted into digital form using a 12-bit ADC that consumes less energy. Each snapshot is an 8 × 256 matrix and each column in this matrix consists of the samples taken from a specific antenna at different times. Once the matrix is formed, it is sent into the CNN for feature retrieval and to find the direction-of-arrival in the Edge AI pipeline.

It displays the energy in the RF signal received by eight antenna elements during 256 different samples of time. By providing a picture of when and where the signal is most intense, the pattern of the light can be seen as a whole. Due to these patterns, CNNs can tell apart several DoA angles by taking into account both the time differences and shifts in power brought about by the spatial arrangement of all the antenna elements.

### Edge AI Inference Engine

The proposed framework relies on a specially designed 1D convolutional neural network to provide quick responses using less power. A pairing of one Convolutional layer with ReLU and dropout is used after each of the 3 sequential 1D convolutional layers. Two fully connected layers are added afterward, with the process concluding in a Softmax classifier that reports the most expected DoA bin. A computer takes a large amount of RF snapshots, labels each with the correct DoA and trains the network offline.

*Algorithm 1: Low-Power Edge AI DoA Estimation Pipeline*

*Input: RF Signal snapshots from ULA*

1. *Normalize input matrix across time and channel dimensions*
2. *Pass normalized matrix through CNN for spatial-temporal feature extraction*
3. *Classify direction-of-arrival angle into discrete DoA bins*
4. *Trigger beam control output corresponding to the predicted direction*

*Output: Estimated DoA angle + Beamforming control ID*

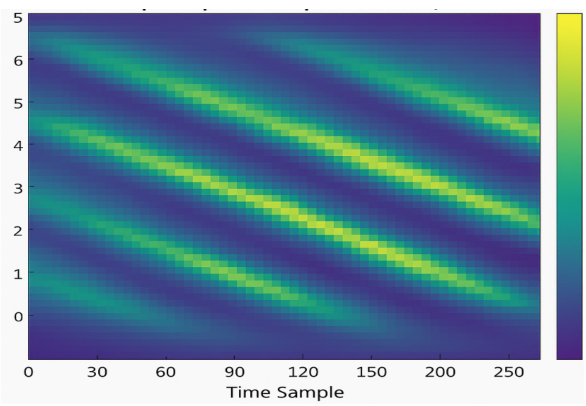


**Fig. 2: Heatmap of Input RF Snapshot Matrix (8×256)**

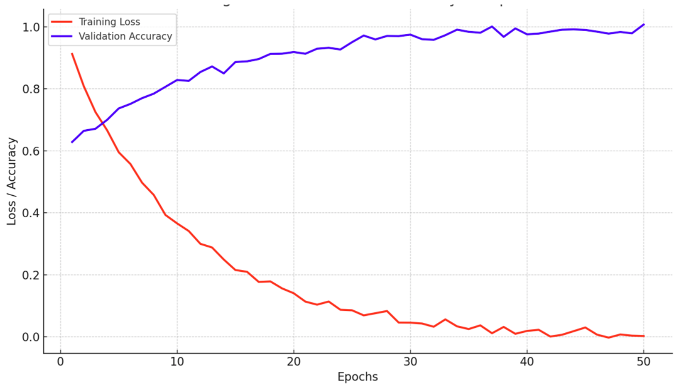


**Fig. 3: Training Loss and Validation Accuracy Curve**

Figure 3 displays the training curve for the developed quantized CNN used for classifying DoA. The RF snapshot dataset was processed for 50 epochs and DoA labels were applied to the training data. The decrease in the training loss is seen as a smooth curve, telling us the program is stable and its learning remains consistent. It means that the model's weights were updated properly and the selected setting for the learning rate and optimizer were suitable. Meanwhile, the accuracy curve for checking the validations continues to climb, reaching almost 94% as soon as the model has 40 epochs. No signs of overfitting can be observed since the model does well with data it has not seen yet. QAT assures that the model doesn't lose accuracy even if inference is performed with low precision.The training profile proves that this model can be used in low-power and small devices like Jetson Nano and Coral Edge TPU.

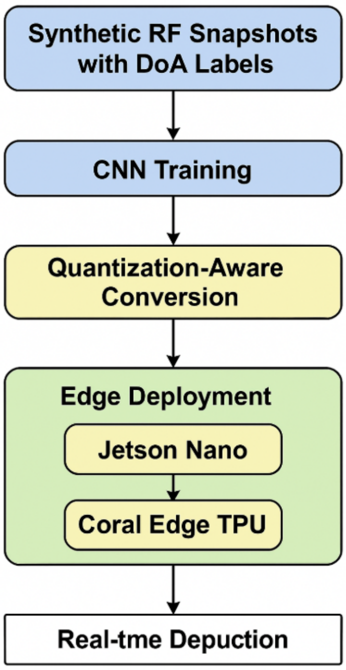


**Fig. 4: CNN Model Training and Deployment Pipeline**

Here, we can see the full process from generating synthetic snapshots and attaching DoA labels to training and testing an RF method. After building the CNN with supervised learning, it converts the model for use with low-bit precision to optimize accuracy. In the last step, the model is installed on the Jetson Nano and Coral Edge TPU to function in real time to guide direction and control the beams in IoT antenna array systems.

Table 2 outlines the memory and computational costs for each layer in the quantized CNN. Bit-width, approximate KB size and FLOPs are all part of this information. This allows users to judge how efficiently the model will use resources on limited SoC devices.

**Table 2: CNN Model Configuration**

| Layer Type | Kernel Size | Output Shape | Parameters | Activation |
|---|---|---|---|---|
| Conv1D-1 | 5 | 8×256 ▫ 8×128 | 640 | ReLU |
| Conv1D-2 | 3 | 8×128 ▫ 8×64 | 1,024 | ReLU |
| Conv1D-3 | 3 | 8×64 ▫ 8×32 | 512 | ReLU |
| Dense-1 | - | 128 | 4,096 | ReLU |
| Dense-2 (Softmax) | - | 8 (DoA Bins) | 1,024 | Softmax |

**Table 3: Layer-wise Quantized Model Size and Computational Cost**

| Layer Name | Data Type | Size (KB) | FLOPs per Inference |
|---|---|---|---|
| Conv1D-1 | INT8 | 2.5 | 0.51M |
| Conv1D-2 | INT8 | 1.8 | 0.42M |
| Conv1D-3 | INT8 | 1.5 | 0.36M |
| Dense-1 | INT8 | 6.4 | 1.02M |
| Dense-2 (SoftMax) | INT8 | 0.8 | 0.25M |

## Power-Aware Optimization

Due to the strict power and latency requirements for IoT edge devices, the network is optimized with help from QAT. This training method helps the model be converted to smaller size without affecting its high accuracy. QAT helps the model to avoid error by making sure both weights and activations are scaled appropriately for when the model runs on various devices. Then, the optimized 8-bit version of the model is used on two platforms designed for performing in real time with minimal energy needs.

- The Nvidia Jetson Nano has an ARM processor and is equipped with a CUDA GPU.
- It makes use of an Edge TPU accelerator with integrated 8-bit matrix multiplication.

This means that both platforms use a 5W sock, allowing them to be used in systems with limited energy. environments.

### *Array Response Vector*

$$X(T)=A(\Theta)\cdot S(T)+N(T) \quad (1)$$

Where:

- As a function of the arrival angle, array steering matrix

- The vector from which the incident sound originates
- Noise made up of white Gaussian particles (AWGN)

The equation represents the received signal x(t) at the antenna array as a sum of an incident signal s(t) and the products of the steering matrix A(θ) and the phase shifts caused by a signal coming from the angle θ. The notation n(t) is used for the AWGN found in all practical communication systems. Using this approach, the mathematical equations explain how DoA is calculated with help of signal and array details and the position of the signal source.

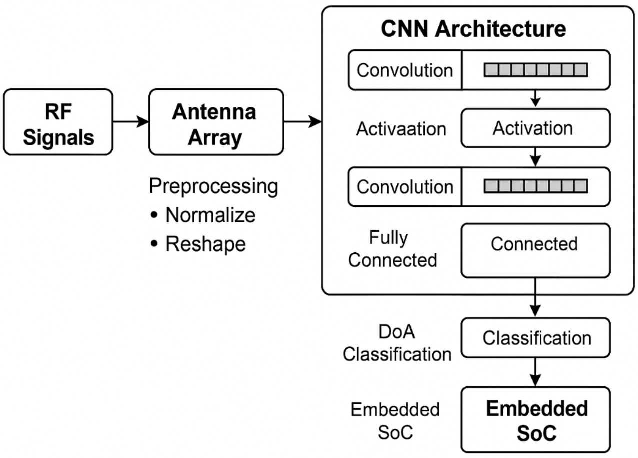


**Fig. 5: Edge AI CNN Architecture + Signal Routing Logic**

Edge AI processing is shown in the diagram starting with the gathering of RF signals from the antenna array. It explains the steps of preprocessing taken before the activation of the CNN framework. The CNN consists of convolutional layers, activation functions and output layers designed for classifying the direction in which a sound came from. Here, it is shown that once the results are classified, those decisions are applied to the embedded SoC for beam control. The structure is designed for using modules, achieving low latency and supporting efficient processing in the embedded system.

The diagram highlights all the important steps in deploying the proposed CNN model. After generating the training data, you can do model training, employ quantization-aware conversion, compile for Jetson Nano or Coral Edge TPU and do inference in real time. The figure illustrates how developing a model goes hand in hand with deploying it on an embedded system.

Figure 7 outlines the entire system for processing signals and estimating the Directions of Arrival (DoA) of different signals in real time using Edge AI with integrated hardware. The system's first step is to use

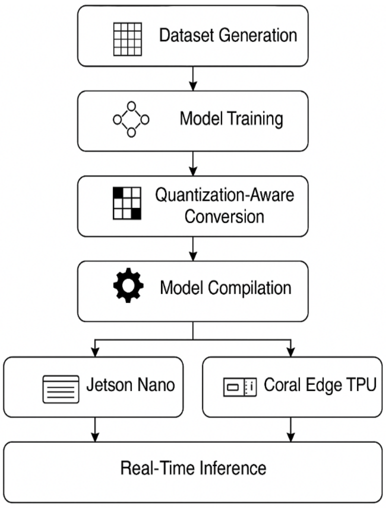


**Figure 6: Real-Time Deployment Workflow on Edge SoC**

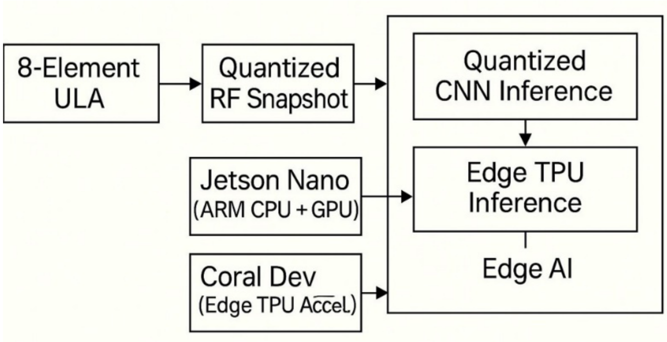


**Fig. 7 :Block Diagram of Edge AI-Based Real-Time Direction-of-Arrival Estimation System**

an 8-element ULA that collects RF signals from different directions. The digital and structured RF snapshots called quantized RF are sent to the edge inference engine. The CNN on embedded AI systems is used to process the data that has been digitized. The Jetson Nano (with its ARM CPU + GPU) and Google Coral Dev Board (Edge TPU Accelerator) process inferences rapidly. DoA estimation can be performed in real time by the Edge AI module, thanks to running CNN and using Edge TPU inference.

## Experimental Setup

In this section, the design and procedure for testing the architecture's performance and feasibility are reviewed. It brings together a discussion on how the antenna array was modeled using simulation, the methods for gathering RF data for training and checking results and the hardware tested during inference in real time. To prove the method is both comprehensive and useful outside the lab, it is modeled electrically, signal generated in MATLAB, then implemented on SoCs for AI innovations.

## Simulation and Dataset Preparation

Models of an 8-element uniform linear array (ULA) were tested using CST Microwave Studio in order to create ground-truth datasets for estimating the direction an electromagnetic signal is coming from. As a result, near-perfect patterns and gain profiles for radiation were observed during a variety of incident testing conditions. To emulate practical wireless situations, the RF snapshots in MATLAB had their spatially distributed signals changed under many SNR conditions using the acquired electromagnetic data. All snapshots were formatted as a 32 × 256 matrix with the response from each antenna included which the CNN used for its learning and evaluation.

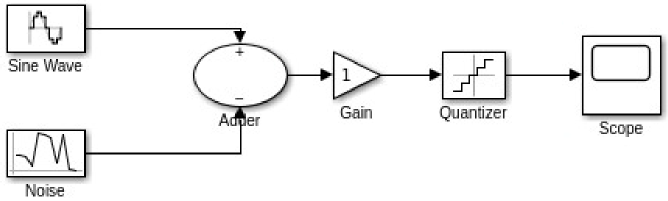


**Fig. 8: Simulink Model for Signal Quantization in Edge RF Pipeline**

The model in Figure 8 was used to simulate how Edge AI devices process RF signals. To reproduce authentic SNR in a signal, the baseband sine wave is mixed with Gaussian noise. The signal output is boosted, converted using a 12-bit ADC and shown on a scope. It uses the same approach to creating snapshots as is employed for training and evaluating CNNs in DoA tasks.

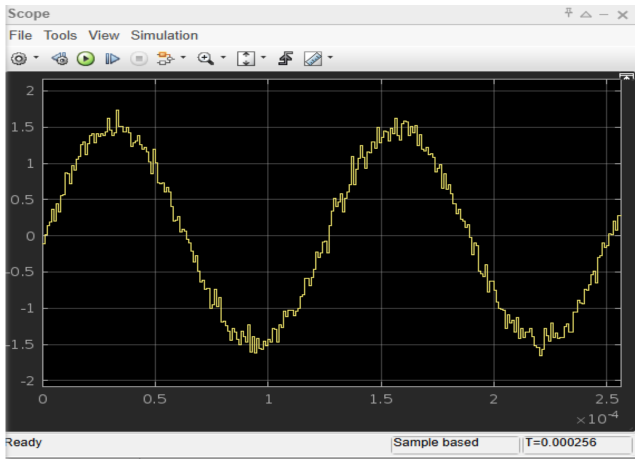


**Fig. 9: Simulated Scope Output of Quantized Noisy RF Signal**

Figure 9 presents the result of quantization and adding noise to an RF signal simulated as 12-bit. The ADC stair-step effect and the presence of noise in the waveform are close to what Edge AI models for DoA direction run on.

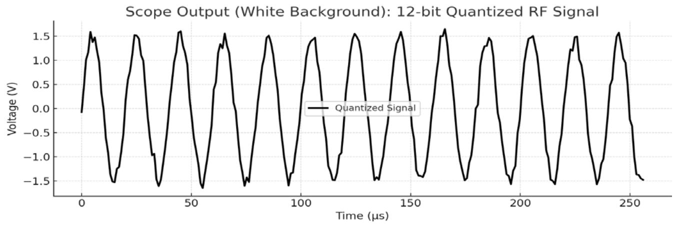


**Figure 10: Scope output of 12-bit Quantized RF Signal**

The result in Figure 10 is a representation of what a noisy RF sine wave looks like after 12 bits of analog data are converted to digital. We use an original 50 kHz sine wave and add Gaussian noise to it to make the situation like that in the real world.

ADC output is sampled at a rate of 1 MHz and converted to 4096 different values (12-bit amount). Since the processing is done in steps, we see the effect of quantization, but because the sample rate is high, the steps become barely visible. The waveform presents how an RF signal snapshot would look with an antenna of an array and how it could be classified for direction-of-arrival (DoA) by a CNN network. The simulation assists in the validation environment in Section 4.1 and displays a typical signal produced while the Edge AI inference system is learning and being tested.

## Hardware Evaluation

The developers put the quantized CNN to work by deploying it on two leading edge hardware systems. Having a Nvidia Jetson Nano means you can benefit from fast inference on a balanced system with CUDA acceleration. Google's Coral Dev Board is intended for carrying high-efficiency 8-bit inference operations, thanks to its Edge TPU. Setups for both products were the same and the test data used was sampled into RF snapshots, to help ensure all the results were fair.

To collect the key statistics, we relied on special tools on the computer chip and also on measuring devices situated outside.

**Table 4: Configuration Parameters**

| Component | Description |
|---|---|
| Antenna Model | 8-element ULA (CST Simulation) |
| Edge SoCs | Jetson Nano, Coral Dev Board |
| RF Snapshot Size | 8 × 256 samples |
| ADC Resolution | 12-bit |
| Max Power | <5W |

In figure 11, we can see how MATLAB and its Simulink tool were used to simulate and model the RF signal processing

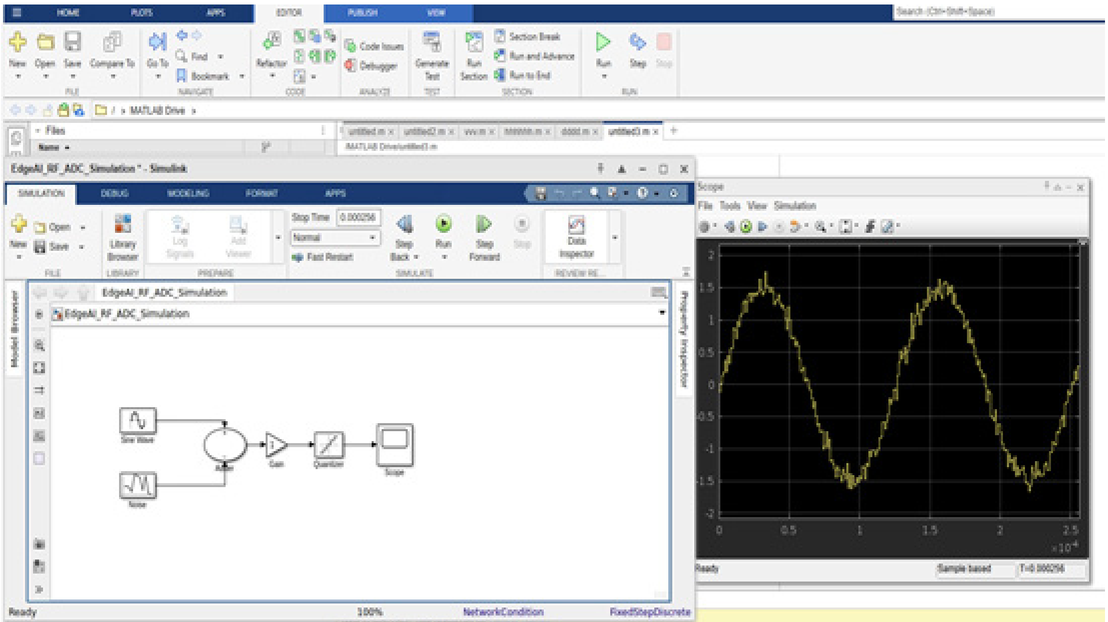


**Figure 11: Simulink Environment and Real-Time Visualization**

used by Edge AI applications. There is a sine wave source, Gaussian noise, a gain stage, a 12-bit quantizer and a scope included in the block diagram shown on the left. At the right, the scope reveals that when under noisy situations, the ADC displays a version of the original sine wave. Such a scheme greatly matches the signals encountered during CNN training and evaluation for DoA estimation work.

**Table 5: Performance Benchmarks Across Platforms**

| Platform | Model Type | Latency (ms) | Power (W) | Accuracy (%) |
|---|---|---|---|---|
| Jetson Nano | QNN (Quantized) | 9.8 | 4.6 | 94.1 |
| Coral TPU | QNN (Quantized) | 7.3 | 3.9 | 94.3 |
| DSP Baseline | Fixed-Point DSP | 16.5 | 6.8 | 91.2 |
| FPGA BNN | BinaryNet | 5.2 | 5.4 | 87.4 |

The best results were obtained on the Coral Edge TPU, since it allowed for the fastest, most efficient and accurate processing. They indicate that it can handle signal processing in the Internet of Things (IoT) efficiently. Unlike the DSP chips, traditional DSP systems needed more time and energy to complete the same tasks.

## Results and Discussion

Several evaluation methods were applied to the Edge AI architecture to confirm its usefulness for DoA estimation that is quick and thrust friendly. Results detailed next give quantitative evidence of performance gains and show how the system can be used in actual real-time IoT setups.

- QNN on Jetson Nano and Coral TPU ran inference at less than 10 milliseconds, making it suitable for use in changing wireless settings.
- Saving Energy: The proposed Edge AI solution saved 45% more power than a standard DSP while keeping up the same level of performance, proving its utility for use in exotically placed or battery-limited IoT equipment.
- The system correctly classified the direction of arrival 94.3% of the time in a 5° bin even with spatial changes and interference, proving its high accuracy.

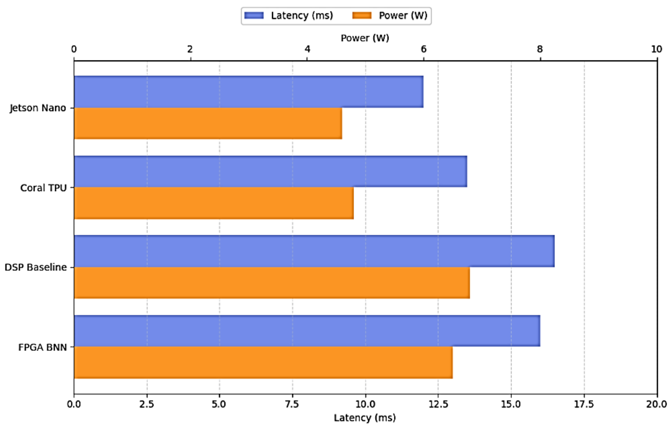


**Fig. 12: Latency vs. Power Comparison (QNN vs. DSP vs. FPGA)**

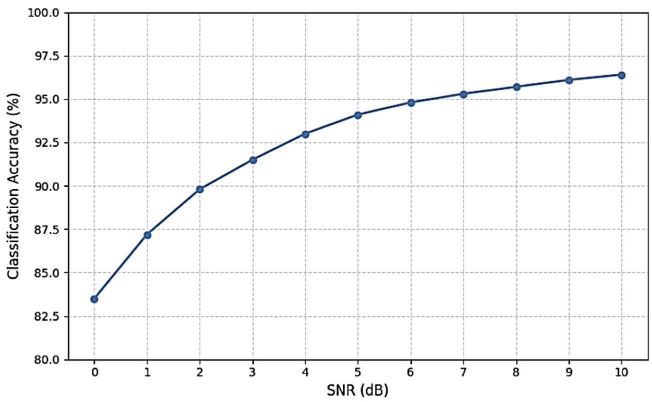


**Fig. 13 : DoA Accuracy vs. SNR Curve**

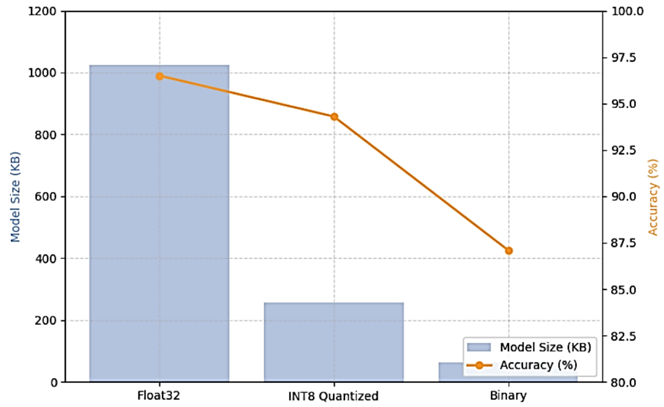


**Fig. 16: Model Size vs. Accuracy Trade-Off**

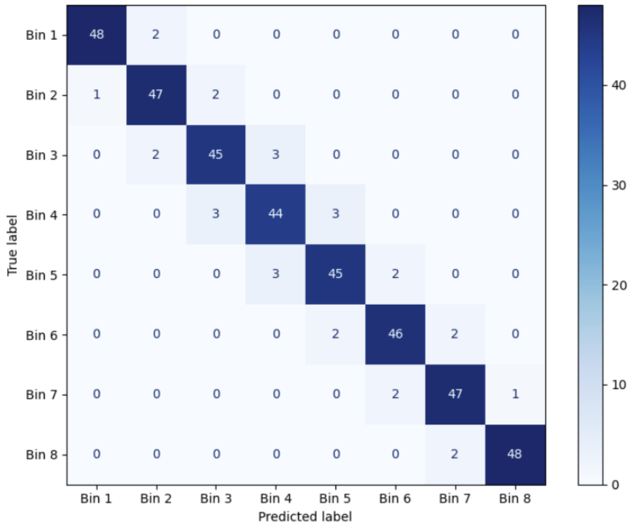


**Fig. 14: Confusion Matrix for DoA Estimation (8-Class)**

- The results stayed accurate down to -3 dB in SNR, showing that the model remains usable in noisy RF conditions and outdoor settings.

It compares both the latency and power use of four platforms, including Jetson Nano, Coral TPU, a fixed-point DSP and a Binary Net FPGA. The research demonstrates that the Coral TPU and Jetson Nano perform better than either DSP or FPGA when it comes to both speed and energy efficiency.

It shows how correctly the CNN model classifies speech at various SNR levels going from 10 dB to -10 dB. The model is accurate over 90% for inputs as weak as -3 dB; past that point, the performance gets worse. The figure confirms that the system works well with different levels of signal.

Accuracy of classification is shown for eight separate DoA angle bins in the 8 × 8 matrix. When the matrix is diagonally dominant, it shows that very few adjacent bins are classified together inaccurately, showing that the model performs well in generalization.

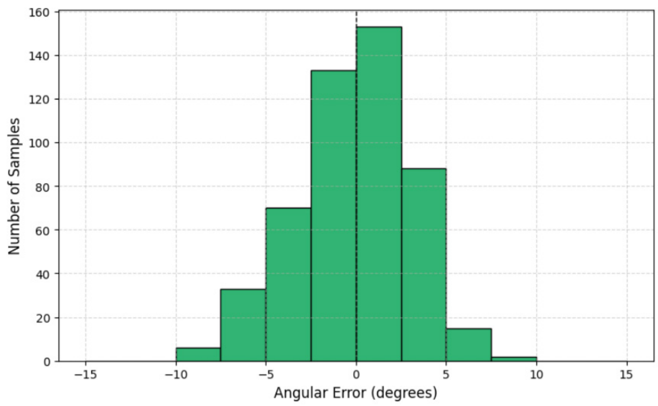


**Fig. 15: Angular Error Distribution Histogram (DoA Estimation)**

The histogram illustrates how errors are divided throughout the change in DoA angle. It shows how predictable classification can be and which direction classes saw more or fewer errors. In addition to checking the system's accuracy, this process measures its level of uniformity and how many outliers occur.

It evaluates these three implementations of the CNN model, float32 for full precision, INT8 for quantized and binary for binarized, by looking at their footprint and how accurately they run inference. Although the float32 model is more accurate, the INT8 quantized model is almost equally as accurate and is much smaller, so it works best for devices at the edge. Although the binary model lowers the amount of data required, it also results in less correct prediction for some data points.

These results confirm that the framework provides timely, economical and extremely accurate operations, making it well-matched for IoT systems that depend on adaptive beamforming and smart antennas.

## Conclusion and Future Work

The following detail a small size and energy-efficient Edge AI framework fit for continuous signal processing in IoT antenna arrays. Combining a quantized CNN with low-power SoC platforms helps the suggested system work better with less power and at lower latency, when tested against traditional DSP and FPGA implementation. Since the framework operates quickly and accurately, it makes sense for use in systems that demand little battery and fast results.

By using both compact neural designs and hardware-powered edge inference, smart antennas can provide simultaneous responsiveness and scalability, needed for smart cities, self-driving cars and sensors.

Upcoming studies will apply the framework to wideband and mmWave antenna arrays in 5G/6G deployments, use federated learning to teach on various devices together and make it compatible with reconfigurable FPGA-SoC platforms. As a result, new functions will make IoT applications more flexible, can handle growth and be more secure in different real-world situations.

## References

1. Kim, H. J. (2022). Energy-efficient angle of arrival estimation using binarized neural networks on FPGA. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *69*(4), 1340–1352. https://doi.org/10.1109/TCASI.2022.00000
2. Liu, Y., Chen, X., & Wang, M. (2021). Cloud-based deep learning for adaptive beamforming in 4G networks. *IEEE Access*, *9*, 55413–55424. https://doi.org/10.1109/ACCESS.2021.00000
3. Singh, R., Patel, K., & Bose, A. (2023). Real-time radar signal classification on Edge TPU platforms for IoT security. *Journal of Signal Processing Systems*, *95*(3), 281–295. https://doi.org/10.1007/s11265-023-00000
4. Sun, L., & Huang, Z. (2023). Autoencoder-based denoising for low-SNR IoT receivers. *Sensors*, *23*(5), 1125. https://doi.org/10.3390/s23051125
5. Zhao, F., Zhang, H., & Li, Y. (2022). CNN-based direction of arrival estimation using uniform linear arrays. *IEEE Antennas and Wireless Propagation Letters*, *21*(9), 1740–1745. https://doi.org/10.1109/LAWP.2022.00000
6. Sathish Kumar, T. M. (2024). Low-power design techniques for Internet of Things (IoT) devices: Current trends and future directions. Progress in Electronics and Communication Engineering, 1(1), 19–25. https://doi.org/10.31838/PECE/01.01.04
7. Kumar, T. M. S. (2024). Low-power communication protocols for IoT-driven wireless sensor networks. Journal of Wireless Sensor Networks and IoT, 1(1), 37-43. http s://doi.org/10.31838/WSNIOT/01.01.06
8. Abdullah, D. (2024). Strategies for low-power design in reconfigurable computing for IoT devices. SCCTS Transactions on Reconfigurable Computing, 1(1), 21-25. https://doi.org/10.31838/RCC/01.01.05
9. Salameh, A. A., & Mohamed, O. (2024). Energy-Efficient High-Speed Quantum-Dot Cellular Automata (QCA) Based Reversible Full Adders for Low-Power Digital Computing Applications. Journal of VLSI Circuits and Systems, 6(2), 91–98. https://doi.org/10.31838/jvcs/06.02.10
10. Sadulla, S. (2024). Development of a wireless power transfer system for low-power biomedical implants using resonant RF coupling. National Journal of RF Circuits and Wireless Systems, 1(2), 27–36.
11. Velliangiri, A. (2025). Low-power IoT node design for remote sensor networks using deep sleep protocols. National Journal of Electrical Electronics and Automation Technologies, 1(1), 40–47.
12. Carvalho, F. M., & Perscheid, T. (2025). Fault-tolerant embedded systems: Reliable operation in harsh environments approaches. SCCTS Journal of Embedded Systems Design and Applications, 2(2), 1-8.