Deep learning and quantization for accurate and efficient multi-target radar inference of moving targets

Nyasha Ernest Mashanda[1], Neil Watson[1], Robert Berndt[2], Mohammed Yunus Abdul Gaffar[3]
[1]Department of Statistical Sciences, Faculty of Science, University of Cape Town, Cape Town, South Africa
[2]Radar and Electronic Warfare, Defence and Security, Council for Scientific and Industrial Research, Pretoria, South Africa
[3]Department of Electrical Engineering, Faculty of Engineering and the Built Environment, University of Cape Town,
Cape Town, South Africa

Article Info

Article history:

Received Jun 14, 2023
Revised Sep 3, 2023
Accepted Sep 12, 2023

Keywords:

Convolutional neural network
Inference
Micro-Doppler
Quantization
Radar

ABSTRACT

Real-time, radar-based human activity and target classification is useful for wide-area ground surveillance. However, the feasibility of deploying deep learning (DL) models in radar-based systems with limited computational resources remains unexplored. This paper investigated the effect of quantization on model throughput and accuracy for deployment in radar systems. A seven-layer residual network was proposed to classify ground-moving targets and achieved a test accuracy of 87.72%. The model was then quantized to 16-bit and 8-bit precision, resulting in a 3.8 times speedup in inference throughput, with less than a 0.4% drop in test and validation accuracy. The results showed that quantization can improve inference throughput with a negligible decrease in target classification accuracy. The increase in throughput and reduction in computational expense that comes with quantization promotes the feasibility of the deployment of DL models in systems with limited computational resources. The findings of this paper hold significant promise for the successful use of quantized models in modern radar systems, while adhering to stringent size, weight and power consumption constraints.

Corresponding Author:

Nyasha Ernest Mashanda
Department of Statistical Sciences, Faculty of Science, University of Cape Town
Rondebosch, Cape Town, South Africa
Email: nyashamash001@gmail.com

1. INTRODUCTION

Human activity and target classification has gained interest in recent years due to its applications in indoor and outdoor surveillance systems for health monitoring [1], border control [2] and security [3]. Different sensor systems have been used for classification, including cameras [4], Lidar [5] and radar [6]. Unlike cameras or Lidar, the performance of a radar sensor is less sensitive to varying weather conditions and different levels of light [7]. Furthermore, radar offers a more extended detection range than optical sensors [6] and can detect targets behind opaque objects [8]. These advantages make radar more suitable for outdoor surveillance systems to curb illegal activities such as poaching, smuggling and livestock theft.

Traditionally researchers have used various techniques to extract pre-defined features for classification from radar data. Examples of such pre-defined features include those related to the physical characteristics of the target [9] and discrete cosine transform coefficients [10]. The features were then used for classification using support vector machines [11] or random forests [12]. This feature estimation and extraction process is highly dependent on human experience and domain knowledge, which makes it susceptible to human error. The advent of deep learning (DL) has allowed an alternative approach to solving the classification problem. DL models enable the automatic identification and extraction of features, which enhances the feasibility of developing a radar-based classification system.

Recent research has shown more complex DL models being used to improve radar-based classification accuracy. In one of the first works, Kim and Moon [13] used a three-layer convolutional neural network (CNN) to achieve a classification accuracy of 90.9% on seven human activities. Subsequently, another investigation [14] developed a convolutional auto-encoder which yielded an accuracy of 94.20% on 12 human activities. Recently, Du *et al.* [15] utilized a ResNet18 model to achieve an accuracy of 95.43% on a six-class human activity dataset. The ResNet18 model had 11 million trainable parameters which makes it computationally expensive to run for inference purposes. This challenge is compounded in an outdoor setting using hardware that has size, weight, power and cost limitations, in addition to stringent latency requirements. Therefore, real-time inference in such environments necessitates careful consideration of a model's computational complexity and associated performance trade-offs.

In image classification, researchers have proposed various ways to compress DL models to reduce their computational requirements, including quantization [16], pruning [17] and knowledge distillation [18]. The deployment of DL models in radar systems requires computational efficiency, especially in systems with limited resources. In surveillance systems, target classification should be done accurately and within a limited specified time interval. The ability to maintain high accuracy is important to achieve practically useful real-time radar-based classification using hardware that has limited computational resources. There has been limited work on optimizing the size of models to improve the inference time for radar-based classification. Thus, the original contribution of this paper is to investigate how quantization affects both the classification accuracy and throughput of a DL model utilized for the purpose of classifying radar data.

2. METHOD

2.1. Problem definition

Moving targets illuminated by radar signals return frequency-modulated signals through the Doppler effect. The Doppler effect is used to estimate the radial velocity of a moving target using (1).

$$f_d \approx \frac{2v \cdot f_c}{c} \qquad (1)$$

where f_d and f_C are the Doppler and carrier frequencies, respectively, v is the target radial velocity, and c is the speed of light.

A target's radial velocity and that of its moving parts can be observed over time in spectrograms as micro-Doppler signatures. The micro-Doppler signatures for targets with different motion patterns, such as walking and running are visually distinct [9]. As a result, CNNs have been used to automatically classify moving targets with different motion patterns based on their Doppler signatures in [13], [19], [20].

2.2. Data collection and pre-processing

A C-band, phased array, pulse-Doppler radar was used for data collection. The radar operated at a center frequency of 5.45 GHz, a pulse repetition frequency of 10 kHz and a pulse bandwidth of 25 MHz. Using the radar, the following activities were measured: one human walking, one human running, two humans walking within 1 meter apart, moving vehicles, clutter and noise with no moving targets, and a 25 cm diameter metallic sphere swinging towards and away from the radar. The ranges of the targets were between 175 meters and 1,400 meters. Baseband in-phase and quadrature samples collected from measuring activities were pulse-compressed [21], beamformed [22] and then decimated to an effective pulse repetition frequency of 714 Hz. Decimation was performed because the Doppler bandwidth provided by 10 kHz was much wider than needed.

A notch filter was then applied to the data to attenuate clutter returns and returns from stationary objects. Spectrograms were created from the filtered data using the short-time Fourier transform (STFT). Three parameters were considered in computing the STFT: window type, overlap and length. In order to strike an optimal balance between frequency and time resolution in the resulting spectrogram, a Hamming window function with a 50% overlap was selected. The STFT procedure commenced by partitioning each signal into smaller windows utilizing the Hamming window function. Following this, the discrete Fourier transform was independently applied to each of these windows. Thus, the STFT of a signal $x[k]$ was given as (2):

$$X_{STFT}[m,n] = \sum_{k=0}^{N-1} x[k]g[k-m]e^{-j2\pi nk/N} \qquad (2)$$

where $g[k]$ denotes an N-point window function, m was the time index, and n was the frequency index. A spectrogram was created by applying the modulus of the STFT result.

$$S[m,n] = |X_{STFT}[m,n]|^2 \quad (3)$$

The selection of the optimal window length, or coherent processing interval (CPI), is a critical aspect of STFT computation. Increasing the CPI with in-phase samples enhances the signal-to-noise ratio (SNR) [23], thereby improving spectrogram signal quality. However, as the coherent processing length encompasses out-of-phase samples, the SNR decreases [23], leading to a decline in spectrogram signal quality. Hence, the peak signal-to-noise ratio (PSNR) from each spectrogram class was used to select the optimal window length.

In order to determine the optimal PSNR, a methodological approach was adopted, involving the random selection of five spectrograms from each class, followed by the computation of the class average PSNR. The findings, as depicted in Figure 1, revealed that, for the majority of classes, the average PSNR exhibited negligible enhancement beyond a CPI of 0.18 seconds, which corresponded to a window size of 128. Consequently, a window size of 128 was deemed appropriate for spectrogram computation, as the adoption of longer CPIs failed to yield substantial SNR improvements that could sufficiently justify the associated increase in computational complexity.

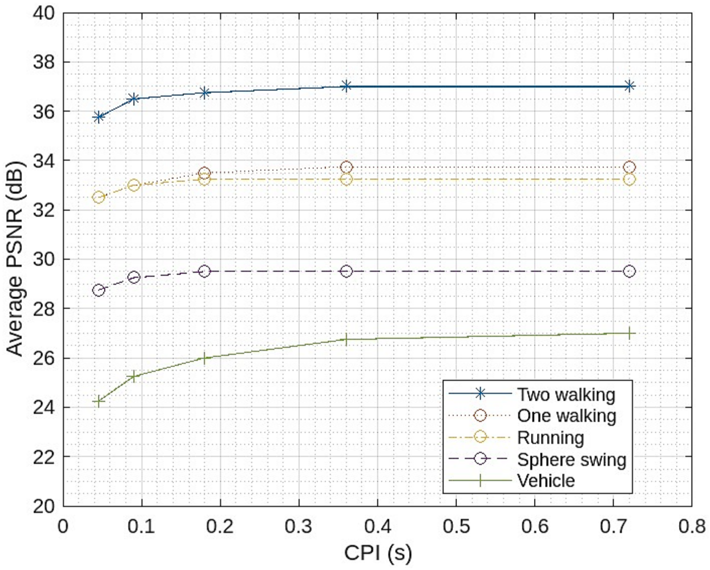

Figure 1. Average PSNR vs. CPI for each activity

A sliding window was used to segment the computed spectrogram, similar to the approach taken in [20]. The overlapped segmentation process serves as a form of data augmentation [24] which promotes enhanced generalization and reduces overfitting in models trained on the data. Each segment was 4 seconds long resulting in spectrograms of size 128 by 45 samples. Figure 2 shows extracted spectrograms of the six classes in the data. It was observed that the micro-Doppler signatures of the different targets were visually unique. Thus, they contained a rich source of information for DL techniques to perform target classification.

A total of 17,939 spectrogram segments were generated from the measured data. The vehicle class had the most segments (4,443) while the sphere swing class was the least represented, with 2,049 examples. The segments were split 70%/15%/15% into the train, validation and test datasets on a per-class basis using the time of day at which they were recorded. Spectrogram segments were divided with respect to the time of recording to avoid leakage of highly correlated spectrograms into the validation and test sets.

2.3. Network architecture

Two CNNs shown in Figure 3 were considered in this study. The first CNN used a standard CNN architecture with convolutional, max-pooling and fully connected layers. Global average pooling (GAP) was applied after the final convolutional layer. The second CNN used residual connections across convolutional layers and GAP after the final convolutional layer.

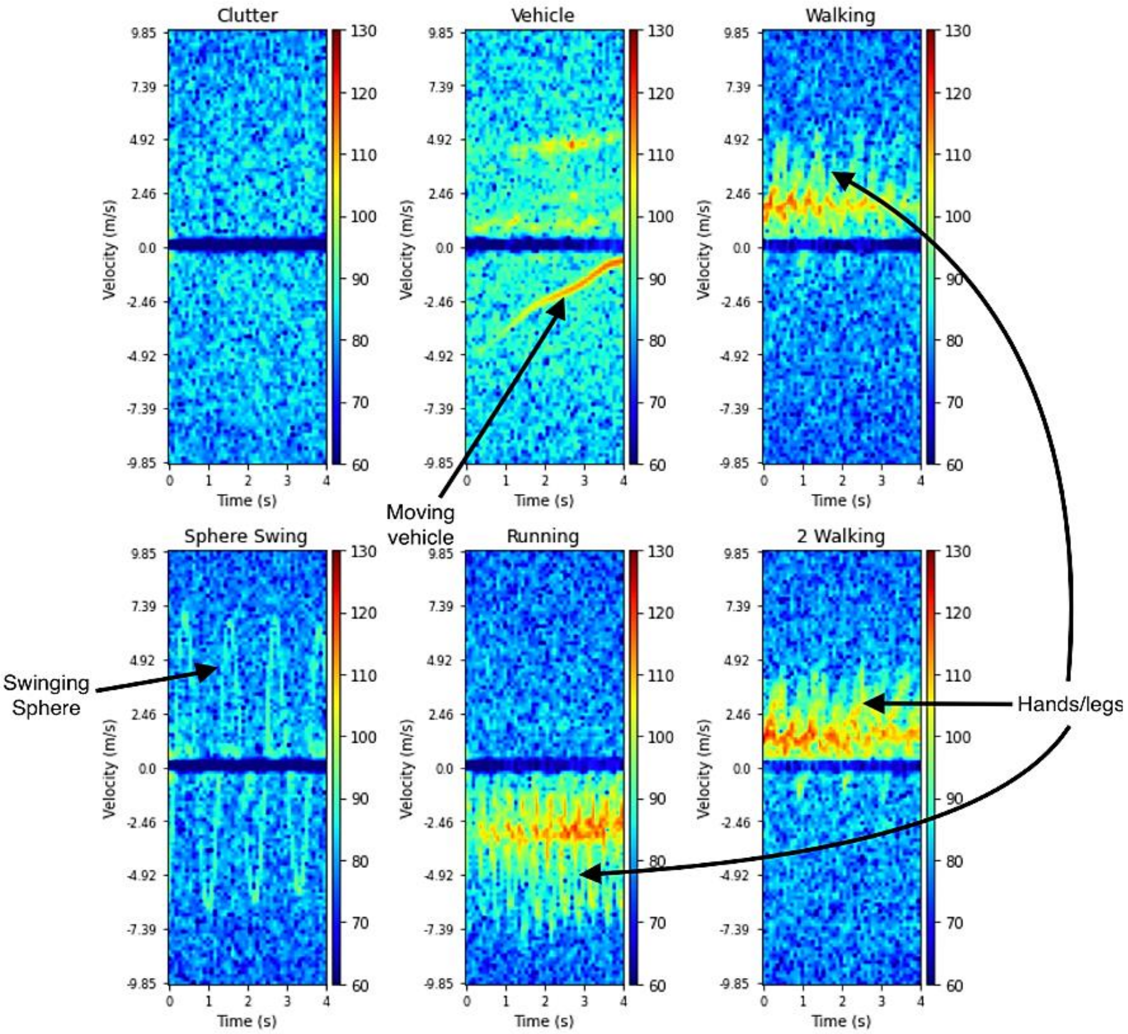

Figure 2. Spectrogram segments of the six classes extracted from spectrograms

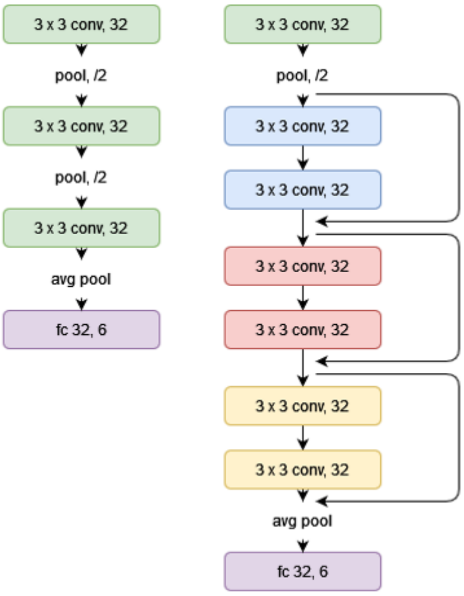

Figure 3. Architecture of model 1 (left) and 2 (right)

The input to each CNN was a 128 by 45 two-dimensional spectrogram. In each convolutional layer, the input was convolved with matrices of weights. Batch normalization [25] was applied to the convolution output to normalize the data to zero mean and unit variance. The rectified linear unit activation function, as described by Nair and Hinton [26], was employed as a non-linear transformation on the normalized output to generate feature maps. Subsequently, max-pooling, a technique utilized for achieving spatial invariance in the context of feature extraction, was applied, as outlined by Scherer *et al.* [27]. This technique enables the identification of features regardless of their specific locations within the input data. Thereafter, GAP was applied to the output of the convolutional and max-pooling layers to mitigate against overfitting and potentially improve the classification performance of the CNN [28]. The output was then flattened before feeding to the fully connected layers to be classified.

The second model had residual connections which skipped two layers (i.e., a residual block) thereby creating a shortcut connection [29]. The shortcut was an element-wise addition of the input to the output of the residual block. This helps to mitigate the vanishing gradient problem which causes accuracy degradation as more layers are added to a network [30].

2.4. Post-training quantization

The model that exhibited the highest accuracy was subjected to post-training quantization (PTQ) [31], a process that reduces the precision for weights and activations within a trained neural network. Lower precision allows efficient use of computational resources and a consequent increase in inference speed [32]. The equation (4) was used to quantize 32-bit floating point (FP32) parameter values to 16-bit floating point (FP16) parameter values:

$$x_q = Half(x) \tag{4}$$

where x is the original FP32 value to be quantized, x_q is the quantized FP16 value, $Half(x)$ is a function that converts the FP32 value x to its FP16 representation using a bit-level transformation. This transformation involves converting the 32-bit single-precision sign, exponent, and significant bits to a 16-bit half-precision format, which has reduced precision compared to FP32 but retains the same bit pattern. Quantization from FP32 to 8-bit integer (INT8) was performed using (5).

$$x_q = clip\left(\lceil \frac{x_f}{\Delta} \rfloor\right) \tag{5}$$

where x_f is a floating-point value, Δ is the step size, $\lfloor\cdot\rceil$ is a function that applies a rounding policy to round rational numbers to representable values in each precision, for example, rounding to integers in INT8 quantization, *clip* is a function that clips outliers that fall outside of the dynamic range of a given precision and x_q is the quantized value.

Equation (6) was used to compute the step size (Δ):

$$\Delta = \frac{q_{range}}{N} \tag{6}$$

where q_{range} is the size of the quantization range and is determined from the distribution of values to be mapped to lower precision. N is the number of representable values in each precision. For example, for an INT8 precision N is 256. Values outside of the quantization range were clipped to the thresholds.

Quantization range setting is an important step in determining the step size, which is crucial for minimizing errors during quantization. There are several methods for determining the quantization range, including max and cross-entropy methods [33]. The max method uses the maximum absolute value of observed floating point values, but it is sensitive to outliers and may cause excessive rounding errors. The cross-entropy method clips outlier values to increase the resolution of inlier values, reducing rounding errors. Therefore, cross-entropy was used for quantization range setting with an input of randomly selected spectrogram segments as a calibration dataset as recommended in [31].

3. RESULTS AND DISCUSSION

All experiments were carried out on a laptop running Ubuntu 18.04, with an Nvidia GTX 1650 GPU and Intel Core i5-9300H (2.40 GHz) processor. The models were trained using PyTorch [34] in FP32. Nvidia TensorRT [35] was used for model optimization and quantization. Model optimization was achieved through layer and tensor fusion, kernel auto-tuning and parallel stream execution. Quantization was applied using TensorRT libraries for FP16 and INT8 quantization.

3.1. Model training and hyper-parameters

Each model applied batch normalization with a momentum of 0.1 and epsilon of 10^{-5} to achieve fast training. Rectified linear unit was used as an activation function. Stochastic gradient descent [36] and adaptive moment estimation (Adam) [37] were considered for optimization using different learning rates from 0.1 to 10^{-5} in orders of 10. Adam led to the best validation accuracy results using a learning rate of 10^{-4} for the proposed models. Furthermore, decreasing Adam's learning rate by 50% after every ten epochs improved the validation accuracy and helped the models converge.

Early stopping [38] was also applied to help prevent the models from over-fitting after observing no improvement in validation loss for 20 epochs. Various batch sizes were also considered in training both models, including 16, 32, 64, 128, and 256. It was found that a batch size of 32 resulted in the highest accuracy. Table 1 summarizes the results obtained from the two CNN models. Model 2 achieved a validation accuracy of 92.90% on the validation data which was 6 percentage points higher than the validation accuracy of model 1. The better performance was likely due to the existence of residual connections, which facilitated the addition of more layers to the network to learn more abstract features without degrading its performance [29].

Table 1. Accuracy results of models 1 and 2

Model	Number of parameters	Training accuracy (%)	Validation accuracy (%)
Model 1	19 206	96.14	86.84
Model 2	56 454	96.69	92.90

Figure 4 shows that model 2 achieved a test accuracy of 95% for all classes, except for the one human walking and two humans walking classes. An analysis of the misclassified spectrograms of the one human walking class revealed that most of the misclassified spectrograms had micro-Doppler signatures with a relatively low SNR. This was due to some of the single human data being measured at the furthest range (1,400 meters) of all recordings. The weak micro-Doppler returns made it difficult to distinguish features between these two classes, resulting in a degradation in accuracy.

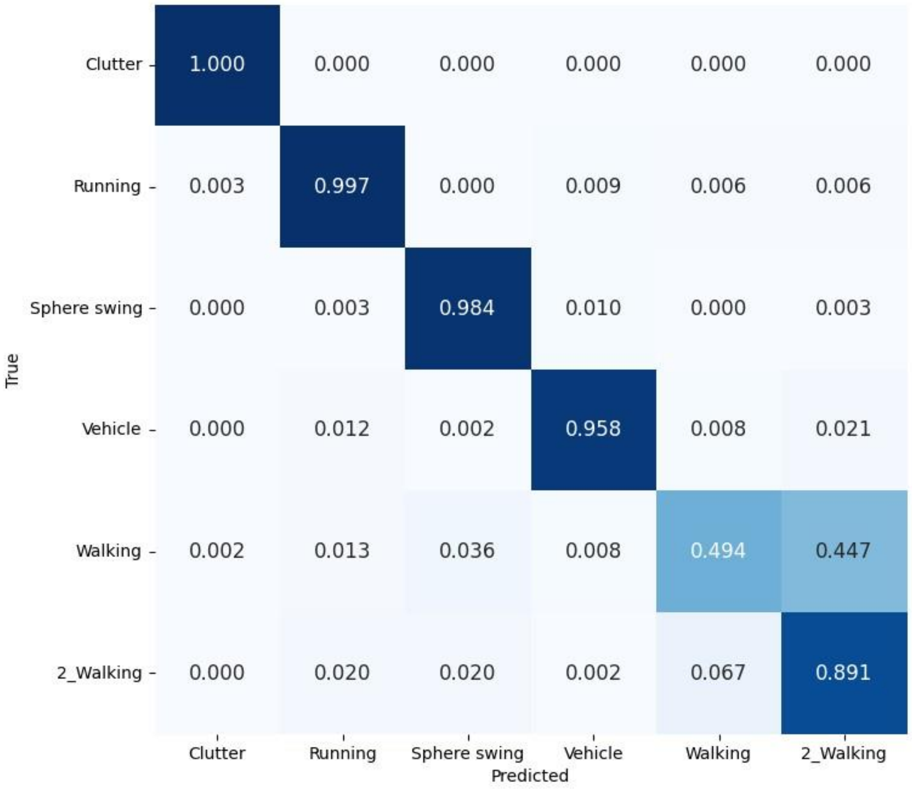

Figure 4. The confusion matrix of model 2 on test data

It is noteworthy that the test accuracy achieved by model 2 in human activity classification was surpassed by the models presented in [14] and [39], both of which attained accuracies exceeding 90%. This observed discrepancy could be attributed to the specific focus of the aforementioned studies on short-range distances, typically under 5 meters, thereby leading to more favorable conditions with high SNR. Consequently, better-quality spectrograms are generated for the classification task.

The two models were further compared using ten-fold cross-validation. In cross-validation, the training and validation data were combined, and the test data was excluded from the process. The combined data was divided into ten folds. The model was trained on nine folds in each training session, using the left-over fold for validation. Model 1 and model 2 achieved 91.32% and 93.02% average cross-validation accuracy, respectively. Therefore, model 2 was chosen as the best model as it showed better predictive performance.

3.2. Effect of quantization on throughput

PTQ was applied to Model 2 based on a PyTorch floating point 32-bit (PFP32) model. Thereafter, the effect of PTQ on throughput and accuracy was investigated. The PFP32 model was optimized by TensorRT and then quantized to FP16 and INT8, resulting in the following models:

a. TFP32 - The TensorRT optimized FP32 model.
b. TFP16 - TensorRT optimized FP16 model.
c. TINT8 - TensorRT optimized INT8 model.

Quantization was performed using a calibration dataset of 84 randomly selected spectrogram segments from each of the six classes to make a total of 504 examples. To measure the throughput of the four models, 1,000 batches of spectrograms were used. Batch sizes of 16, 32, 64, 128 and 256 were considered. The formula used to calculate throughput was as (7):

$$Throughtput = \frac{N}{T}\ (spectrograms/s) \tag{7}$$

where N is the total number of classified spectrograms and T is the total time taken to complete inference.

The results in Figure 5 show an increase in throughput as the batch size increased. This was because bigger batch sizes allow more spectrograms to be processed in parallel, leading to higher throughput. However, the throughput plateaued for a batch size greater than 128 due to limitations on the number of parallel processes that can be run using the available computational resources. Figure 5 also shows that the PFP32 model had the lowest throughput among the four models. The PFP32's highest throughput was 5,000 spectrograms/s using a batch size of 256. Quantizing the TensorRT optimized model to INT8 precision resulted in the highest speed gain of 3.8 times from 10,000 spectrograms/s to 38,000 spectrograms/s using a batch size of 256.

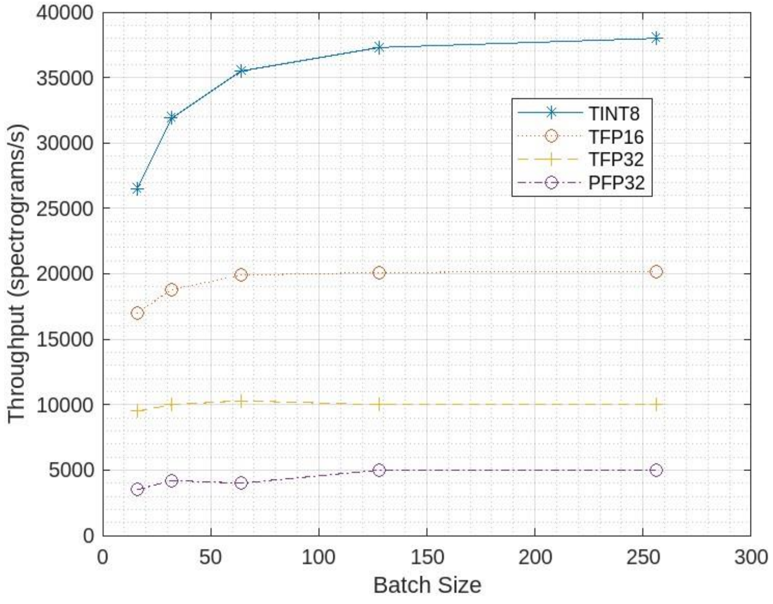

Figure 5. Data throughput vs. batch size and quantization precision

It is worth noting that the INT8 model throughput is lower compared to the 5.7 to 7.3 times found in [32] using MobileNet on the CIFAR10 and FashionMNIST datasets. This difference can be explained by the fact that different models manifest distinct throughout improvements due to their architectural characteristics and input data as shown in [32].

Furthermore, it is pertinent to consider the potential for enhancing model throughput by the adoption of specialized processors optimized for low-precision wide vector arithmetic. Remarkably, the findings in [16] elucidate a speedup of up to 10 times when employing the Qualcomm digital signal processor equipped with hexagon vector extensions (HVX). Such dedicated processors tailored to low-precision computations thus hold promise for bolstering model throughput in practical deployment scenarios.

3.3. Effect of quantization on accuracy

After calculating throughput, the accuracy of the four models was compared on the validation and test sets to assess how quantization affected the performance of the model. The accuracy results are summarized in Table 2. Quantization resulted in a negligible decrease in accuracy in both the validation and test data. INT8 quantization had a larger accuracy drop compared to FP16 in both datasets. The maximum percentage drop was 0.33% on the validation data and 0.38% on the test data. The percentage drop in accuracy was higher for the INT8 model than the FP16 model because of an increase in information loss when moving from 32-bit to 8-bit precision (as opposed to 16-bit precision). Furthermore, the drop in accuracy was lower than the 1% experienced by MobileNet in [32] which is plausibly due to the different model architectures and input datasets considered.

Table 2. Comparison results of model 2 with different precisions at a batch size of 256

Model	Validation accuracy (%)	Test accuracy (%)	Latency (*µs/spectrogram)*	Throughput (*spectrogram/s)*
PFP32	92.90	87.72	200	5 000
TFP32	92.90	87.72	100	10 000
TFP16	92.75	87.70	48.8	20 500
TINT8	92.57	87.34	26.3	38 000

4. CONCLUSION

This study presented a residual network model tailored specifically for outdoor radar-based human activity and target classification using spectrograms. Through the process of quantization, the model’s inference speed experienced a notable enhancement, while maintaining a negligible loss in accuracy. The achieved accuracy of 87.72% on the test data exemplifies the model’s efficacy in accurate activity identification, underscoring its potential value in practical applications. Although challenges were observed in the misclassification of the human walking class, the model’s overall capacity to discriminate between human activity and non-activity is an encouraging outcome.

Future investigations should prioritize addressing the misclassification issue to fortify the model’s performance within this specific class, thereby elevating its suitability for real-world deployment in outdoor surveillance scenarios. Notably, quantization from 32-bits to 8-bits yielded compelling results, manifesting a significant quadrupling of throughput, while incurring less than 0.4% reduction in accuracy. Moreover, the potential for further speed enhancements through the utilization of specialized hardware emphasizes the model’s feasibility and efficiency in practical deployment contexts.

In essence, this research makes an important contribution by demonstrating that radar-based classification models in outdoor surveillance systems can be effectively optimized via quantization, thereby enhancing inference speed while maintaining high accuracy. Such a finding holds profound implications for the feasibility and viability of deploying radar-based models within these critical settings. The findings of this paper, together with the potential use of specialized inference hardware, hold significant promise for the successful use of DL techniques in modern radar systems, while adhering to stringent size, weight and power consumption constraints.

ACKNOWLEDGEMENTS

This research was supported by the Council for Industrial and Scientific Research (CSIR), Pretoria, South Africa and funded by the Mandela Rhodes Scholarship.

REFERENCES

[1] M. G. Amin, Y. D. Zhang, F. Ahmad, and K. C. D. Ho, “Radar signal processing for elderly fall detection: the future for in-home monitoring,” *IEEE Signal Processing Magazine*, vol. 33, no. 2, pp. 71–80, Mar. 2016, doi: 10.1109/MSP.2015.2502784.

[2] S. Z. Gurbuz, U. Kaynak, B. Ozkan, O. C. Kocaman, F. Kiyici, and B. Tekeli, “Design study of a short-range airborne UAV radar for human monitoring,” in *2014 48th Asilomar Conference on Signals, Systems and Computers*, Nov. 2014, pp. 568–572, doi: 10.1109/ACSSC.2014.7094509.

[3] F. Fioranelli, M. Ritchie, and H. Griffiths, “Classification of unarmed/armed personnel using the NetRad multistatic radar for micro-Doppler and singular value decomposition features,” *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 9, pp. 1933–1937, Sep. 2015, doi: 10.1109/LGRS.2015.2439393.

[4] X. Yang and Y. Tian, “Super normal vector for human activity recognition with depth cameras,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 5, pp. 1028–1039, May 2017, doi: 10.1109/TPAMI.2016.2565479.

[5] F. Luo, S. Poslad, and E. Bodanese, “Temporal convolutional networks for multiperson activity recognition using a 2-D lidar,” *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7432–7442, Aug. 2020, doi: 10.1109/JIOT.2020.2984544.

[6] D. Tahmoush, J. Silvious, and J. Clark, “An UGS radar with micro-Doppler capabilities for wide area persistent surveillance,” in *Radar Sensor Technology XIV*, Apr. 2010, vol. 7669, pp. 26–36, doi: 10.1117/12.848233.

[7] A. S. Mohammed, A. Amamou, F. K. Ayevide, S. Kelouwani, K. Agbossou, and N. Zioui, “The perception system of intelligent ground vehicles in all weather conditions: A systematic literature review,” *Sensors*, vol. 20, no. 22, Nov. 2020, doi: 10.3390/s20226532.

[8] L. M. Frazier, “Surveillance through walls and other opaque materials,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 11, no. 10, pp. 6–9, 1996, doi: 10.1109/62.538794.

[9] Y. Kim, S. Ha, and J. Kwon, “Human detection using Doppler radar based on physical characteristics of targets,” *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 2, pp. 289–293, Feb. 2015, doi: 10.1109/LGRS.2014.2336231.

[10] P. Molchanov, J. Astola, K. Egiazarian, and A. Totsky, “Ground moving target classification by using DCT coefficients extracted from micro-Doppler radar signatures and artificial neuron network,” in *2011 Microwaves, Radar and Remote Sensing Symposium*, Aug. 2011, pp. 173–176, doi: 10.1109/MRRS.2011.6053628.

[11] Y. Kim and H. Ling, “Human activity classification based on micro-Doppler signatures using a support vector machine,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 5, pp. 1328–1337, May 2009, doi: 10.1109/TGRS.2009.2012849.

[12] M. Ritchie, M. Ash, Q. Chen, and K. Chetty, “Through wall radar classification of human micro-Doppler using singular value decomposition analysis,” *Sensors*, vol. 16, no. 9, Aug. 2016, doi: 10.3390/s16091401.

[13] Y. Kim and T. Moon, “Human detection and activity classification based on micro-Doppler signatures using deep convolutional neural networks,” *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 1, pp. 8–12, Jan. 2016, doi: 10.1109/LGRS.2015.2491329.

[14] M. S. Seyfioglu, A. M. Ozbayoglu, and S. Z. Gurbuz, “Deep convolutional autoencoder for radar-based classification of similar aided and unaided human activities,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 4, pp. 1709–1723, Aug. 2018, doi: 10.1109/TAES.2018.2799758.

[15] H. Du, Y. He, and T. Jin, “Transfer learning for human activities classification using micro-Doppler spectrograms,” in *2018 IEEE International Conference on Computational Electromagnetics (ICCEM)*, Mar. 2018, pp. 1–3, doi: 10.1109/COMPEM.2018.8496654.

[16] R. Krishnamoorthi, “Quantizing deep convolutional networks for efficient inference: a whitepaper,” *arXiv:1806.08342*, Jun. 2018.

[17] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, “Pruning convolutional neural networks for resource efficient inference,” *arXiv:1611.06440*, Nov. 2016.

[18] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” *arXiv:1503.02531*, Mar. 2015.

[19] R. P. Trommel, R. I. A. Harmanny, L. Cifola, and J. N. Driessen, “Multi-target human gait classification using deep convolutional neural networks on micro-Doppler spectrograms,” in *2016 European Radar Conference (EuRAD)*, 2016, pp. 81–84.

[20] Y. He, Y. Yang, Y. Lang, D. Huang, X. Jing, and C. Hou, “Deep learning based human activity classification in radar micro-Doppler image,” in *2018 15th European Radar Conference (EuRAD)*, 2018, pp. 230–233, doi: 10.23919/EuRAD.2018.8546615.

[21] J. R. Klauder, A. C. Price, S. Darlington, and W. J. Albersheim, “The theory and design of chirp radars,” *Bell System Technical Journal*, vol. 39, no. 4, pp. 745–808, Jul. 1960, doi: 10.1002/j.1538-7305.1960.tb03942.x.

[22] B. D. Van Veen and K. M. Buckley, “Beamforming: a versatile approach to spatial filtering,” *IEEE ASSP Magazine*, vol. 5, no. 2, pp. 4–24, Apr. 1988, doi: 10.1109/53.665.

[23] R. Awadhiya and R. Vehmas, “Analyzing the effective coherent integration time for space surveillance radar processing,” in *2021 IEEE Radar Conference (RadarConf21)*, May 2021, pp. 1–6, doi: 10.1109/RadarConf2147009.2021.9455335.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[25] S. Ioffe and C. Szegedy, “Batch normalization: accelerating deep network training by reducing internal covariate shift,” in *International conference on machine learning*, 2015, pp. 448–456.

[26] V. Nair and G. E. Hinton, “Rectified linear units improve restricted Boltzmann machines,” in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814, doi: 10.5555/3104322.3104425.

[27] D. Scherer, A. Müller, and S. Behnke, “Evaluation of pooling operations in convolutional architectures¨ for object recognition,” in *International conference on artificial neural networks*, vol. 6354, K. Diamantaras, W. Duch, and L. S. Iliadis, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 92–101.

[28] M. Lin, Q. Chen, and S. Yan, “Network in network,” *arXiv:1312.4400*, Dec. 2013.

[29] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778, doi: 10.48550/arXiv.1512.03385.

[30] W. Liu and K. Zeng, “SparseNet: a sparse DenseNet for image classification,” *arXiv:1804.05340*, Apr. 2018.

[31] M. Nagel, M. Fournarakis, R. A. Amjad, Y. Bondarenko, M. van Baalen, and T. Blankevoort, “A white paper on neural network quantization,” *arXiv:2106.08295*, Jun. 2021.

[32] K. Paupamah, S. James, and R. Klein, “Quantisation and pruning for neural network compression and regularisation,” in *2020 International SAUPEC/RobMech/PRASA Conference*, 2020, pp. 1–6, doi: 10.1109/SAUPEC/RobMech/PRASA48453.2020.9041096.

[33] H. Wu, P. Judd, X. Zhang, M. Isaev, and P. Micikevicius, “Integer quantization for deep learning inference: principles and empirical evaluation,” *arXiv:2004.09602*, Apr. 2020.

[34] A. Paszke *et al.*, “Pytorch: an imperative style, high-performance deep learning library,” *Advances in neural information processing systems*, vol. 32, 2019.

[35] E. Jeong, J. Kim, S. Tan, J. Lee, and S. Ha, "Deep learning inference parallelization on heterogeneous processors with TensorRT," *IEEE Embedded Systems Letters*, vol. 14, no. 1, pp. 15–18, Mar. 2022, doi: 10.1109/LES.2021.3087707.
[36] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv:1609.04747*, 2016.
[37] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," *arXiv:1412.6980*, 2014.
[38] L. Prechelt, "Early stopping-but when?," *Early stopping-but when?. In Neural Networks: Tricks of the trade*, pp. 55–69, 2002.
[39] J. Zhu, H. Chen, and W. Ye, "A hybrid CNN–LSTM network for the classification of human activities based on micro-Doppler radar," *IEEE Access*, vol. 8, pp. 24713–24720, 2020, doi: 10.1109/ACCESS.2020.2971064.

BIOGRAPHIES OF AUTHORS

Nyasha Ernest Mashanda received his B.Sc. degree in electrical engineering and a M.Sc. degree in data science from the University of Cape Town, South Africa in 2019 and 2021 respectively. He was a machine learning engineer at Peralex between 2020 and 2021, specializing in computer vision. In 2022, he joined Quantium as a data scientist. His research interests are radar signal processing and machine learning. He can be contacted at nyashamash001@gmail.com.

Neil Watson received his B.Sc. and B.Sc. Hons (mathematical statistics) degrees from Nelson Mandela University, before completing his Masters in operational research for development in the Department of Statistical Sciences at UCT in 2011. He completed his Postgraduate Certificate in Education at UNISA while teaching at Bishops Diocesan College in Cape Town from 2011 to 2012, after which he joined the Department of Statistical Sciences at UCT in 2013 as a lecturer. His research interests are in the fields of data science, decision support systems and sports statistics. He can be contacted at nm.watson@uct.ac.za.

Robert Berndt completed a B.Eng. electronic engineering degree at the University of Pretoria, South Africa, in 2007. Since 2008 he has been with the Council for Industrial and Scientific Research (CSIR), Pretoria, South Africa, as a radar signals and systems analyst. His main fields of interest are radar signal processing, machine learning, and target recognition with a specific focus on micro-Doppler techniques. While at the CSIR he has completed his B.Eng. (Honours) in electronic engineering at the University of Pretoria (2009) and is currently enrolled at the University of Cape Town, South Africa, where he is studying towards a Ph.D. degree in electrical engineering. He can be contacted at rberndt@csir.co.za.

Mohammed Yunus Abdul Gaffar received the B.Sc. Eng. and M.Sc. Eng. degrees in electronic engineering from the University of Natal in 2002 and 2003 respectively, and subsequently received his Ph.D. from the University of Cape Town. He was with the Council for Scientific and Industrial Research (CSIR) from 2003 to 2016, where he was active in the fields of inverse synthetic aperture radar and radar detection of ground-moving objects. In 2016, he joined the University of Cape Town as a senior lecturer. His research interests are in the fields of short-range radar systems and radar signal processing. He can be contacted at yunus.abdulgaffar@uct.ac.za.