Quantifying Degradations of Convolutional Neural Networks in Space Environments

Emily Altland, Jonathan Castellanos, Joshua Detwiler, Paolo Fermin, Raquel Ferrá, Conor Kelly, Casey Latoski, Tiffany Ma, Thomas Maher, Julia Mahon Kuzin, Ali Mohammadian, Abdelrahman Said Abdalla, W. Chris Headley, and Alan J. Michaels
Virginia Tech Hume Center for National Security and Technology
{emilya99, jonatc5, joshde, paolofermin, rferra99, conork, clatoski, mstffy, tomm13, juliam8, alim2, asaid97, cheadley, ajm}@vt.edu

Abstract— **Advances in machine learning applications for image processing, natural language processing, and direct ingestion of radio frequency signals continue to accelerate. Less attention, however, has been paid to the resilience of these machine learning algorithms when implemented on real hardware and subjected to unintentional and/or malicious errors during execution, such as those occurring from space-based single event upsets (SEU). This paper presents a series of results quantifying the rate and level of performance degradation that occurs when convolutional neural nets (CNNs) are subjected to selected bit errors in single-precision number representations. This paper provides results that are conditioned upon ten different error case events to isolate the impacts showing that CNN performance can be gradually degraded or reduced to random guessing based on where errors arise. The degradations are then translated into expected operational lifetimes for each of four CNNs when deployed to space radiation environments. The discussion also provides a foundation for ongoing research that enhances the overall resilience of neural net architectures and implementations in space under both random and malicious error events, offering significant improvements over current implementations. Future work to extend these CNN resilience evaluations, conditioned upon architectural design elements and well-known error correction methods, is also introduced.**

I. Introduction

CNNs are becoming increasingly prominent in industry today as a machine learning method to perform classification, prediction, and other tasks. The greatest success of neural networks has been their ability to far surpass other machine learning techniques in computer vision tasks [1,2]; however, they have also been successful at natural language processing [3] and RF signal processing [4-6]. Neural networks are now being used by almost every sector of industry and academia and are being given increasingly challenging tasks. From detecting pedestrians in driverless cars [7] to being used in the atmospheric sciences to predict weather patterns [8], neural network applications are likely to expand. In particular, this research seeks to quantify the suitability of extending CNN-based algorithms to harsh space environments, where cosmic radiation induces performance degrading hardware malfunctions in computational elements and stored neural network parameters during operation.

One common example of hardware errors is those encountered as single event upsets (SEUs). These SEUs are most often represented by bit flips and can be caused by harsh environmental factors as demonstrated in [9,10]. There are many examples of environmentally induced SEUs in modern technology [11,12]. Examples include satellites exposed to cosmic radiation [13] and imaging applications in harsh environments like after the Fukushima disaster [14]. There is also evidence of adversarial attacks being employed to cause randomly distributed bit flips, such as row hammer attacks on protected memory cells [15], leading to a desire to understand where effort should be invested to improve implementation resilience. When these errors are accumulated over time, the consequences of such SEUs can be detrimental to implemented algorithms. This paper seeks to quantify those rates of performance degradation as a function of the number of SEUs.

The widespread use of neural networks on mainstream platforms raises questions as to how their performance is affected by SEUs. The robustness of CNN architectures against SEUs was previously reported by [9,10]. However, the performance of the neural network did not degrade proportionally to the amount of SEUs induced in memory. Instead, the degradation was bimodal in nature, suggesting that there may be multiple competing degradation factors at play. This paper seeks to establish a more in-depth evaluation of SEU impacts in the context of CNN-based image recognition and RF modulation classification algorithms. This work is conditioned upon the actual subset(s) of bits where SEUs occur – done via controlled errors occurring in different locations of the floating point number representation (sign, exponent, and mantissa). The results confirm that exponent and sign bit errors are the predominant contributors to CNN degradations, quickly reducing performance to random guessing, while mantissa errors more gradually degrade the CNN-implemented algorithm’s performance. To validate the breadth of this intuitive observation, performance evaluations were performed for a collection of three open source machine vision neural nets (ResNet50, InceptionV3, and MobileNetV2,) and a custom RF machine learning signal/modulation classifier [4].

978-1-7281-0048-7/19/$31.00 ©2019 IEEE

This paper ultimately aims to create a performance baseline for impacts of space radiation on CNNs. This baseline focuses on number representation and will support future work on CNN architectural components and error correction methods to deploy more resilient CNN-based algorithms. The simulation framework for the family of neural nets considered in this study is introduced in Section II, followed by experimental results and analysis enabling isolation of the SEU-induced errors and performance limits in Section III. Conclusions and an introduction to ongoing work to improve CNN resilience is introduced in Section IV.

II. CNN Simulation Framework

A common test environment was built around the number representation of coefficients used within a CNN for these experiments, focusing on standard IEEE single data formats as used in most ground-based processing architectures. A system diagram of the analysis framework is shown in Figure 1, beginning with the trained CNN, where all model training is assumed to be completed a priori. An ensemble of errored models are then created based on the selection of randomly selected seed values for the SEU emulation process and a variable quantity of accumulated SEU events. The final stage in the model is to evaluate degraded model performance via the use of a common validation dataset, and then compare those to the baseline performance of an unerrored model. As such, the observed degradations are taken as a population of random error events over the ensemble of random error cases.

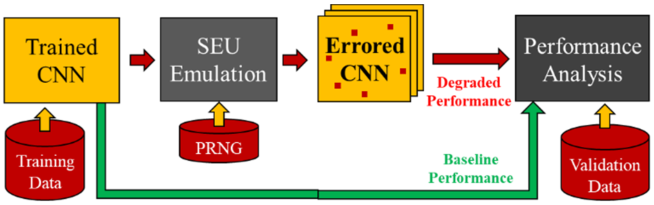

Figure 1. Top-level simulation framework for quantifying degradations caused by randomly occurring SEUs.

The primary focus of this paper is to create a quantifiable segmentation of error impacts found in [10], which resulted in an observable bimodal performance degradation when taken as uniformly random SEU events across any coefficient within the CNN and any bit within the coefficient. At low numbers of SEUs, the accuracy is not greatly affected; however, once the quantity of SEUs reaches a threshold, the CNN accuracy begins to devolve to random guessing. This behavior is different from the originally expected behavior, which was a steady, monotonic, degradation of accuracy in some proportion to the accumulated SEUs. To further investigate the cause of this bimodal behavior, ten distinct targeted SEU scenarios were defined. In each case, random weights or biases were chosen without preference; however, errors were constrained to a specific range of bits in the coefficient data structure. The bit indices and corresponding SEU cases are captured in Figure 2. An expectation in these distinct cases is that the errors will become increasingly impactful as they propagate from the LSB to more significant bits, with what are likely quantifiable differences as errors impact exponent and sign bits. Separation of the exponent and sign bits into distinct cases from the mantissa bits offers isolation of their impacts within the CNN structure; these will be shown later to be critical factors when neural net performance degrades to random guessing.

Cases 1-7 target bits in the mantissa of the IEEE single numeric data structure:

Case 1: LSBs 11 down to 0 are isolated for SEU impacts
Case 2: Bits 17 down to 12 are isolated for SEU impacts
Case 3: Bits 22 down to 18 are isolated for SEU impacts
Case 4: SEUs only occur in bit 19
Case 5: SEUs only occur in bit 20
Case 6: SEUs only occur in bit 21
Case 7: SEUs only occur in bit 22

Cases 8-10 target bits in the exponent and sign bits of the IEEE single numeric data structure:

Case 8: LSBs 3 down to 0 of the exponent are isolated
Case 9: MSBs 7 down to 4 of the exponent are isolated
Case 10: Sign bit of the word is isolated for SEU events

An additional goal of this paper is to identify the systematic resilience characteristics of CNNs in general. Therefore, four different CNNs were evaluated; all but one of these uses the ImageNet dataset [16], making the validation dataset common, and thus the degraded performance a function only of the network architecture and structure. Further, the normalization of errors as a fraction of the total number of CNN parameters is consistent with CNN deployment of such CNNs in a hazardous environment that accounts for memory die area.

A. CNN Models Evaluated

CNN architectures provide inherently sparse interactions within the convolutional layers. As opposed to fully connected layers where each output is connected to all the inputs, each

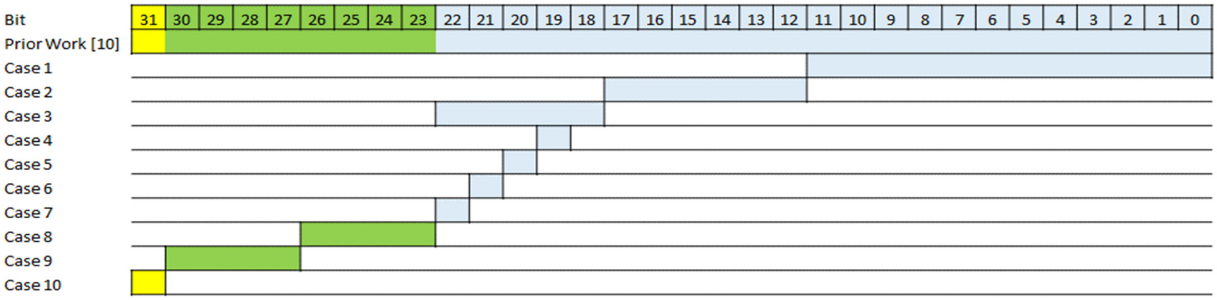

Figure 2. Targeted SEU case descriptions.

output to a convolutional layer is only connected to a small number of inputs. These convolutional layers require less operations to calculate the outputs and results in fewer parameters than fully connected layers. Each one of the previously mentioned neural networks employ convolutional layers in conjunction with other tactics to achieve a faster and accurate CNN. The CNNs evaluated in this paper use an input and output layer, with many hidden layers connecting the two. Each layer works together to extract features from the image that assists it with classifying the output into a desired class. Four distinct CNNs were evaluated: (1) ResNet50 [17], (2) InceptionV3 [18,19], (3) MobileNetV2, [20-22] and (4) a custom RF machine learning (RFML) modulation classifier [4]. These algorithms were chosen to show the impacts of degradations on low, medium, and high performance image recognition neural networks, respectively, as well as at least one data point for a CNN-based solution for RF signals.

1) ResNet50

The ResNet family of image recognition CNNs was created by Microsoft Research in 2015. ResNet50, as well as the collection of ResNet variants, was designed to alleviate the training necessary for neural networks that process a relatively deep number of layers. This neural net formulates layers as learning residual functions with respect to the layer inputs, as opposed to learning unreferenced functions. ResNet50 is one amongst an ensemble of residual neural networks which managed to achieve a 3.57% error rate on the ImageNet test set. This resulted in winning first place for the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2015 classification task [17]. ResNet50 is a 50 layer residual network, uses primarily 3x3 filters, down samples with CNN layers with a stride of 2, and has a global average pooling layer combined with a 1000-way fully connected layer at the end.

2) InceptionV3

InceptionV3 is the 2016 iteration of the network developed by researchers at Google which introduced inception layers [18,19]. Inception layers are convolutional layers, but with several different filter sizes in a single layer. The output of the different filters is concatenated together before being passed as an input to the next layer. This relies on the principle that large (5x5, 7x7) convolutional filters use a disproportionately high amount of resources, and can be replaced by several smaller filters. This allows it to perform well even under strict memory and computational constraints. The architecture of the network consists of six convolutional layers of 3x3 filters, several inception layers of varying dimensions, and finally a pooling layer, linear, and 1000-way softmax. With 42 layers in total, InceptionV3 achieves greater accuracy than previous neural nets while also using fewer resources.

3) MobileNetV2

The MobileNet family of computer vision models was developed in 2017 by Google [20-22]. MobileNetV2 was released in April of 2018. It was designed to be suitable for environments with less compute power, such as in mobile and embedded-based vision applications. Within MobileNet, the developers introduced two hyper-parameters, a width multiplier and a resolution multiplier, which efficiently trade between latency and accuracy and allow the model builder to choose the appropriately sized model for their application. MobileNetV2 has three convolutional layers, which includes a 1x1 expansion convolutional layer, a 3x3 depth-wise convolution, and a 1x1 projection layer. Using depth-wise separable convolutions, MobileNet can be used to build light weight and highly efficient deep neural networks.

4) RFML Modulation Classifier

A custom convolutional neural network was built to classify radio signals as either QPSK, BPSK, QAM16, or QAM64 frequency modulation. The architecture for this network was chosen based off prior research to evaluate convolutional layers and filters for RF classification. The RF classifier consists of 3 convolutional layers with 45 filters and a kernel size of 3x3. The input layer takes input of size 1024x2 representing radio signals generated using the GNU radio module. After the 3 convolutional layers, the data neural network's architecture is flattened to a 1-dimensional shape and consists of a dense layer of 128 nodes, a dropout layer, and final output layer of 4 nodes each representing one of the four radio frequency modulations. The baseline accuracy, tested against new radio signal data generated from GNU radio modules, is 93.2%. There are a total of 11.8M weights and biases and 0 non-trainable parameters.

5) Model Comparison

These four CNNs represent vastly different architectures where ResNet introduced an identity shortcut connection that can skip one or more layers, MobileNetV2 uses linear bottlenecks between the layers and shortcut connections between the bottlenecks, InceptionV3 uses *inception* layers, and the RFML modulation classifier processes quadrature vectors (1024x2 input layer), has 5 layers, and was built to classify radio signals. MobileNetV2 is built for mobile and embedded applications where resources are constrained as described in [23]. Due to these varying architectures, there is a vast difference in the number of parameters each CNN uses, as shown in Table 1.

Table 1. Comparison of resources for evaluated CNNs.

	# of Layers	# of Parameters	# of Bias	# of Weights
MobileNetV2	28	3.5 M	18,956	3,520,028
Inception V3	42	24 M	18,712	23,833,072
ResNet50	50	26 M	54,120	25,582,592
RFML Modulation Classifier	5	12 M	267	11,833,847

B. Experimental Setup

The same framework as described in Section II for inducing SEUs and testing accuracy was carried out for each CNN architecture. Each error case was evaluated over a logarithmic scaling of accumulated SEUs representing the percentage of parameters that experience an SEU. All induced errors are applied uniformly randomly, leading to a likelihood that SEUs may trigger errors in the same bit once error rates reach 1%-10%. For each SEU percentage, a Monte Carlo ensemble of 100 trials was simulated, with the average CNN accuracy of the ensemble as the degraded performance.

An important point for the experimental setup is that the models are pre-trained and thus do not include performing the actual training processes, so no SEU-induced errors are emulated during training. The testing datasets are common among all degraded models within the errored ensemble. Goals for this experimental setup, and related future work, are to isolate where errors occur and then translate those errors into techniques that may be used to improve the overall resilience of CNNs deployed to space platforms.

III. Experimental Results

Details of the performance degradations across all experimental cases are presented for each CNN in Section IIIA, followed by a cumulative analysis that identifies common patterns in performance degradations across the neural nets in Section IIIB. Finally, these results are extrapolated to estimate the expected lifetimes of neural nets when deployed [unshielded] in four different operational scenarios having different prevalence of SEU flux densities.

A. Performance Degradations of Errored CNNs

Results of each error cases for the four CNNs are provided in Figure 3; panel (a) provides the results for MobileNetV2, panel (b) provides the results for InceptionV3, panel (c) provides the results for ResNet50, and panel (d) provides the results for the RF machine learning modulation classifier. Common to each of these plots is a logarithmic x-axis that represents the fraction of total CNN parameters receiving SEU errors and a linear y-axis that summarizes the average degraded performance level of the resulting ensemble of errored CNNs. An additional guide is provided for the assumed performance of an image/modulation classifier that has devolved to random guessing (calculated as the inverse of the total number of classes).

B. Analysis of Results

In evaluating the performance degradations occurring in SEU-induced errored CNNs, the analysis is broken into three distinct categories, consistent with the IEEE floating point word structure: errors isolated to the mantissa (cases 1-7), errors isolated to the exponent (cases 8-9), and errors in the sign bit (case 10). Overall behavior patterns between each of the CNNs align well with these categories.

1) Errors in Mantissa (Cases 1-7)

Similarly to prior work [10] that evaluated the impact of reduced precision CNN implementations, Cases 1 and 2 show minimal if any observable impact to the overall performance to the chosen neural net for all CNNs evaluated. As such, the bottom 18 bits of the IEEE single representation may either be eliminated to reduce on-orbit memory requirements or perhaps re-allocated to improve the resilience of the actual stored coefficients. Case 3, which is effectively a uniform distortion arising from the mantissa MSBs, begins to demonstrate a concrete reduction in classification performance levels. Closer inspection of the mantissa MSBs in Cases 4 through 7 demonstrates that the true impacts are isolated in most cases to the mantissa MSB itself (Case 7), although the other bits (Cases 4-6) do have some visible impacts at higher accumulated SEUs. In all cases of the mantissa errors, it is important to note that the implicit leading 1 value in the IEEE single format is not subject to error, which is truly a function of coefficient scaling. In all cases, the errors to the mantissa elements of CNN parameters appear to result in gradual degradations in performance that accumulate over time.

2) Errors in Exponent (Cases 8-9)

Both test cases for the exponent bits yield marked degradations in classification performance, with the exponent LSBs (Case 8) yielding a lower, and apparently bounded, reduction in performance, while the MSBs (Case 9) rapidly force the CNN classification performance to degrade to the point of random guessing. General SEU accumulation levels, where classification performance has degraded by 20% from unerrored performance, are in the range of 10^{-4}-10^{-2} percent of parameters. An interesting note about Case 8 is that for 1-10 percent of weight errors, the accuracies of many cases increase. While this may be an artifact of the limited Monte Carlo runs, another potential explanation is accumulated SEUs correcting previously errored bits.

One observation of the CNN parameters, and an anticipated side effect of batch normalization processes that are used during training, is that the exponent fields are sparsely used. Based on histograms of the raw coefficients, the exponents are limited to 1-2 bits of effective use, suggesting that the MSBs are almost completely unnecessary, offering better performance if fixed point parameter representations are used instead of floating point, matching observations from [25]. This observation also suggests the need to focus attention of future resiliency methods towards addressing the scaling of CNN parameters such that SEU impacts do not result in cascading degradations in the nonlinear CNN structures.

3) Errors in the Sign Bit (Case 10)

For the final case of sign bit errors (Case 10), a similar, yet less pronounced, impact of the exponent becomes apparent. Accumulated SEUs in the sign bit do rapidly lead to a reduction in the overall classification performance, with the larger two CNNs losing classification accuracy gradually (demonstrating greater resilience), while the smaller CNNs converge to an accuracy substantially higher than random guessing when sufficiently many sign bits have errors.

4) Summary of Error Cases

While the results of this analysis yield intuitive results (i.e., that the coefficient scaling and sign bits have the greatest impact on CNN classification performance when exposed to SEUs, while mantissa errors lead to gradual degradations), the isolation and quantification of the performance degradations offer a foundation for future research that will seek to improve implementation resilience. In addition, the observed gradual degradations of Cases 1-7 and rapid de-evolution to random guessing in Cases 8-10 offers a concrete explanation as to the bifurcation in classification results from [10] when SEUs arbitrarily impact bits in memory, which is comparable to a superposition of error cases between Cases 1-7 and Cases 8-10. Finally, the relative brittleness of the trained CNN solutions, is apparent since a random accumulation of 100-500

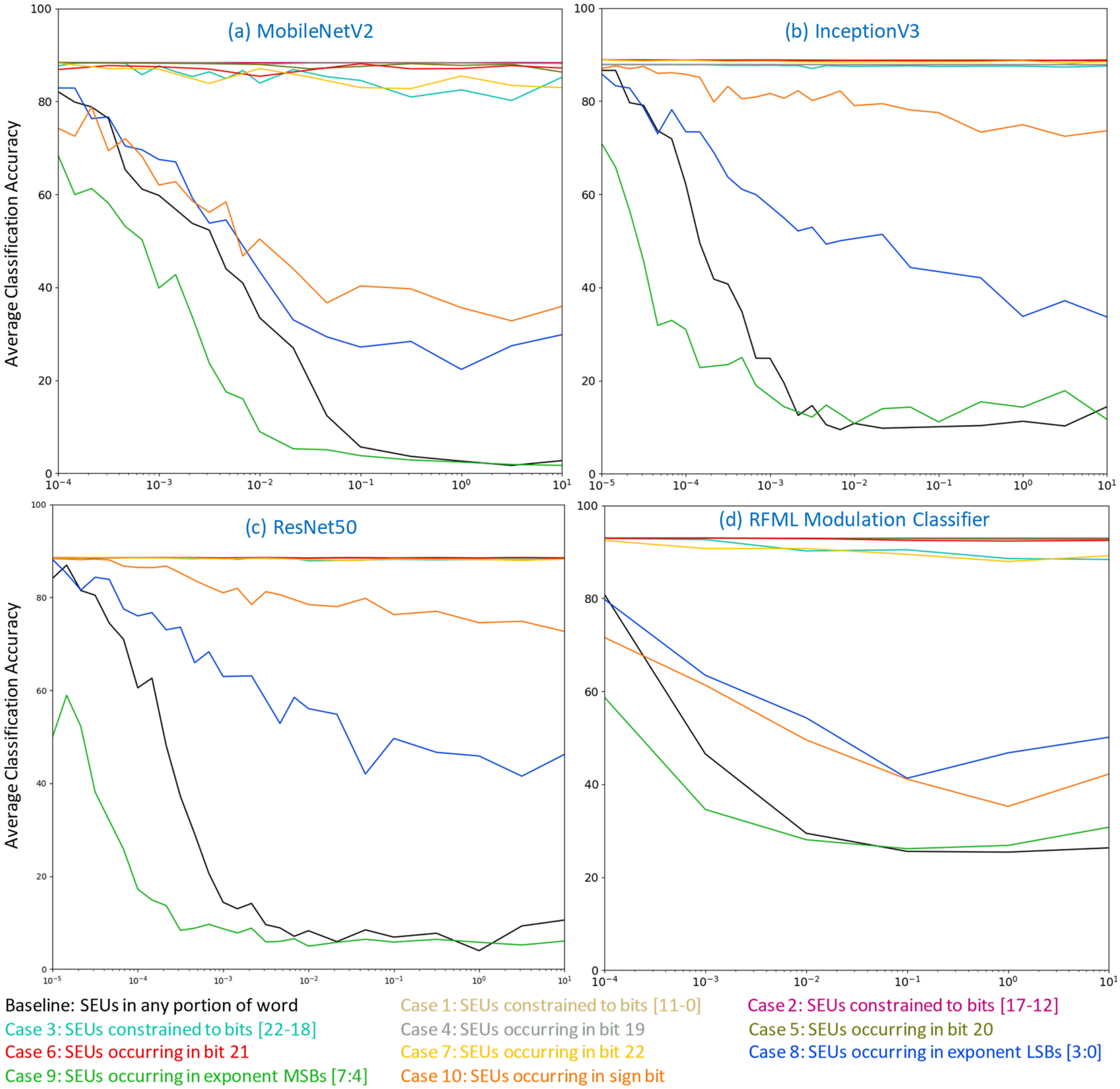

Figure 3. Summary performance degradations for four CNN architectures when exposed to controlled SEU conditions. Panels include performance degradations for (a) MobileNetV2, (b) InceptionV3, (c) ResNet50, and (d) an RF machine learning modulation classifier. Degraded performance shown as an average classification accuracy (y-axis) taken over a logarithmically scaled accumulation of SEU errors (x-axis) as a percentage of total CNN parameters.

errors (predominately driven by exponent field SEUs, which comprise 1/4th of the single precision data format) can destroy the performance of a chosen CNN classification algorithm.

An interesting feature of Figure 3 is that when high levels of SEUs are induced, the performance increases marginally. This effect is particularly significant with the RF results. Cases 8-10, where bits are flipped in the exponent and sign bit, show this trend most clearly. In effect, the degraded CNNs appear to retain some amount of its learned behaviors.

Comparing the results of smaller networks to larger networks, the small sample size suggests that smaller networks appear to be more resilient. This could be because the average distance (additional layers) that an error propagates in a small neural network is less, meaning that the average error in a smaller neural network is less detrimental than that in a larger neural network. Another explanation could simply be that induced more errors in larger neural networks, because the amount of SEUs induced is proportional to the neural network size. The RF network degraded at a slower rate than the other architectures. One explanation for this could be that the RF neural network classifies input into 4 categories rather than 1,000, like the other architectures. This means that even if an SEU changes probability of each category, for an arbitrary input, since there are only 3 other categories, rather than 999, the same category is far more likely to still have the highest probability for the RF neural network than any other in this study. Regardless, more study will be required to isolate and quantify the source of these effects.

C. *Operational Lifetimes for Chosen Neural Nets*

The final question for the analysis is to help translate the observed performance degradations into anticipated lifetimes for CNN algorithms when deployed in a space environment. In each of four cases: (a) Earth surface [25], (b) Low-Earth Orbit (LEO) [26,27], (c) Moon surface [28,29], and Mars surface [30,31], the average expected lifetime of the CNN when exposed to the prevailing level of cosmic radiation is provided. No allowances are made in these cases for shielding, scrubbing, or any other resilience enhancements. Overall results match expectations from previous sections, where the "expected" results may be approximated as a weighted superposition of Cases {1,2,3,8,9,10}, with Cases {8,9,10} dominating performance degradations. Under the assumptions of identical memory architectures using 16nm SRAM cells [32] and Poisson random arrival processes for the SEUs (i.e., a uniform probability of any CNN parameter receiving an error) [33], a direct comparison of the normalized fractions of errors in each of the CNNs may be made. SEU error rates per model and environment are shown in Table 2.

Table 2. Calculated SEU-induced error rates (errors/second) for four CNNs in space environments [25-31].

	Earth	LEO	Moon	Mars
MobileNetV2	$3.16 \cdot 10^{-8}$	$1.8 \cdot 10^{-4}$	$8.77 \cdot 10^{-7}$	8.77
ResNet50	$2.28 \cdot 10^{-7}$	$1.30 \cdot 10^{-3}$	$6.48 \cdot 10^{-6}$	64.81
InceptionV3	$2.11 \cdot 10^{-7}$	0.0012	$5.85 \cdot 10^{-6}$	58.5
RFML Modulation Classifier	$1.05 \cdot 10^{-7}$	$5.8 \cdot 10^{-4}$	$2.92 \cdot 10^{-6}$	29.2

For these results, an "unstable" model is assumed to be one that performs at less than 80% of the peak performance. Given that the SEU arrival rates are scalar multiples for each environment (i.e., the lifetimes for different environments will look similar, yet with shifted x-axes), a focus on LEO orbits is provided in Figure 4, which ultimately shows that the MobileNetV2 image classification CNN is more resilient that the other two ImageNet approaches. Performance results for the RF modulation classification algorithm degrade at a similar rate to MobileNetV2.

Given the performance levels shown in Figure 4, the better CNNs evaluated have a lifetime in LEO, estimated as 80% of baseline performance, on the order of 20 hours. Given the assumed Poisson arrival process, lifetimes in the other radiation environments may be viewed as re-scaled x-axis views of Figure 4, yielding an Earth surface lifetime of 13.2 years, moon surface lifetime of 173.6 days, and a Mars surface lifetime of 1.50 seconds. Further, the MobileNetV2 neural net architecture performs best for image classification in space environments, despite being the smallest overall CNN and having a lower baseline performance. Finally, the RFML modulation classification CNN appears slightly less resilient than MobileNetV2. In all cases, the converging performance level for highly accumulated SEU cases is close to, yet slightly positive, when compared to random guessing, which is 0.1% for ImageNet and 25% for RF modulation classification, suggesting that some learned behaviors are retained independent of the number of errors.

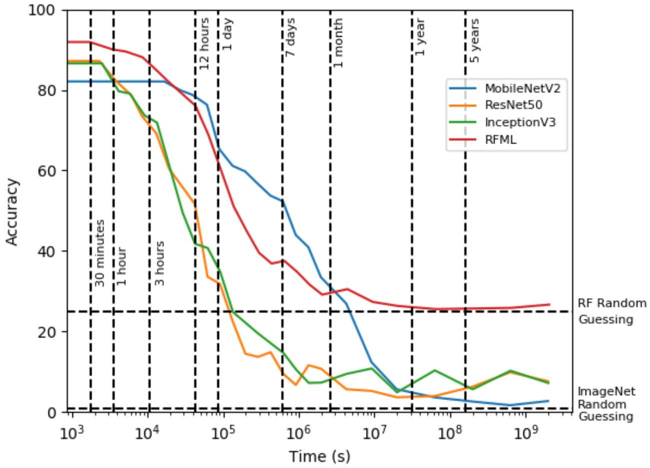

Figure 4: Expected lifetimes and performance levels of four different CNN architectures exposed to LEO radiation levels.

IV. Conclusions and Future Research

This paper has evaluated and quantified the performance degradations arising from deployment of four different CNN-based classification (four image classification and one RF signal classification) algorithms when subjected to a space radiation environment and the associated single event upsets that occur in stored memory parameters. The results isolated specific portions of the standard IEEE single data format used for most CNN parameters, identifying the impact of SEUs when they occur in different portions of the mantissa, exponent, and sign bit. SEUs induced in the 18 LSBs of the mantissa had little to no effect on CNN accuracy, which suggests these bits may be manipulated in ways to protect the memory stored in the MSBs. These conclusions confirm prior work [10] observations that hypothesized bifurcating SEU impact results from Monte Carlo runs as being caused by the underlying number representation.

Further, this paper has created a quantified foundation for ongoing work to improve CNN resiliency in space environments, both as a function of data format and CNN architecture. Specifically, error detection and correction algorithms such as triple mode redundancy (TMR) and parity bit checks could be implemented. Parity bit checks would allow a user to monitor the 'health' of a CNN and stop making predictions when a certain threshold of SEUs have been induced. This would be an ideal application to protect neural networks deployed in harsh environments. Future work will also evaluate the adversarial angle of CNN implementations for the scenario of targeted bit errors, both as a function of number representation and architecture, with anticipated degradations quicker than these shown in emulations of a natural environment.

V. ACKNOWLEDGEMENTS

The authors wish to thank the Naval Surface Warfare Center in Crane, IN, for their sponsorship of this work under grant N00174-18-1-0005. Thank you also to VT Advanced Research Computing and NVidia for support with GPU resources.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” 2012.

[2] S. Loussaief and A. Abdelkrim, "Machine learning framework for image classification," 7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications, pp. 58-61, 2016.

[3] A. Graves, A. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 6645–6649, 2013.

[4] S. C. Hauser, W. C. Headley, A. J. Michaels, “Signal detection effects on deep neural networks utilizing raw IQ for modulation classification,” IEEE Military Communications Conference, pp. 121-127, 2017.

[5] S. Peng, et al., "Modulation Classification Based on Signal Constellation Diagrams and Deep Learning," in IEEE Trans. on Neural Networks and Learning Systems, vol. 30, no. 3, pp. 718-727, March 2019.

[6] L. J. Wong, W. C. Headley, and A. J. Michaels, “Estimation of transmitter I/Q imbalance using convolutional neural networks,” IEEE 8th Annual Computing and Communication Workshop and Conference, pp. 948-955, 2018.

[7] D. Tome, F. Monti, L. Baroffio, L. Bondi, M. Tagliasacchi, and S. Tubaro, “Deep convolutional neural networks for pedestrian detection,” Journal of Image Communication, Vol. 47, pp. 482-489, 2016.

[8] S. Ghosh, et al., “Weather Data Mining using Artificial Neural Network,” 2011 IEEE Recent Advances in Intelligent Computational Systems, Trivandrum, India, pp. 192-195, 2011.

[9] A.P. Arechiga and A.J. Michaels "The effect of weight errors on neural networks" IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC) pp. 190-196 Jan. 2018.

[10] A. P. Arechiga, “Sensitivity of Feedforward Neural Networks to Harsh Computing Environments,” M.S. thesis, Virginia Tech, 2018.

[11] R. Mastipuram and E. Wee, “Soft errors’ impact on system reliability,” Electrical Design News, pp. 69-74, Sep 30, 2004.

[12] J. Saleh, D. Hastings, and D. Newman, “Weaving time into system architecture: satellite cost per operational day and optimal design lifetime,” Acta Astronautica. 54. 413-431, 2003.

[13] S. E. Hoyos, H. D. R. Evans and E. Daly, "From Satellite ion flux data to SEU rate estimation," in IEEE Transactions on Nuclear Science, vol. 51, no. 5, pp. 2927-2935, Oct. 2004.

[14] D. R. Case, "The application of real-time image processing to self-guided vehicles operating in a hostile environment," Fifth Int. Conference on Image Processing and its Applications, pp. 389-393, 1995.

[15] Y. Kim, et al., “Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors,” ACM/IEEE 41st International Symposium on Computer Architecture, pp. 361-372, 2014.

[16] O. Russakovsky, et al. “ImageNet Large Scale Visual Recognition Challenge,” International Journal of Computer Vision, 2015.

[17] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, 2016.

[18] C. Szegedy et al., "Going deeper with convolutions," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1-9, 2015.

[19] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818-2826, 2016.

[20] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 4510 - 4520.

[21] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” 2017.

[22] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, pp. 4510 - 4520, 2018.

[23] A. Canziani, E. Culurciello, and A. Paszke, “An Analysis of Deep Neural Network Models for Practical Applications,” 2017.

[24] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan. “Deep learning with limited numerical precision” in Proceedings of the 32nd Int. Conference on Machine Learning, Vol. 37, 1737-1746, 2015.

[25] “Measurement and reporting of alpha particle and terrestrial cosmic ray-induced soft errors in semiconductor devices.” Technical Report JESD89A, October 2006.

[26] I. Getselev, S. Rumin, N. Sobolevsky, M. Ufimtsev, and M. Podzolko. “Absorbed dose of secondary neutrons from galactic cosmic rays inside the international space station.” Advances in Space Research, 34(6):1429–1432, 2004.

[27] H. Koshiishi, H. Matsumoto, A. Chishiki, T. Goka, and T. Omodaka. “Evaluation of the neutron radiation environment inside the International Space Station based on the Bonner Ball Neutron Detector experiment.” Radiation Measurements, 42(9):1510–1520, October 2007.

[28] R. E. Lingenfelter, E. H. Canfield, and W. N. Hess. “The lunar neutron flux.” Journal of Geophysical Research, 66(9):2665–2671, 1961.

[29] R. E. Lingenfelter, E. H. Canfield, and V. E. Hampel. “The lunar neutron flux revisited.” Earth and Planetary Science Letters, 16(3):355–369, 1972.

[30] J. Khler, et al. “Measurements of the neutron spectrum on the Martian surface with MSL/RAD.” Journal of Geophysical Research: Planets, 119(3):594–603, 2014.

[31] J.W. Wilson, et al., “Deep space environments for human exploration.” Advances in Space Research, 34(6):1281–1287, January 2004. ISSN 02731177.

[32] C. Slayman. “Soft error trends and mitigation techniques in memory devices.” Annual Reliability and Maintainability Symposium, pp. 1–5, 2011.

[33] N. A. Dodds, et al., “The contribution of low-energy protons to the total on-orbit SEU rate,” OSTI.GOV, 01-Aug-2015.