Analyzing the Resilience of Convolutional Neural Networks Implemented on GPUs: Alexnet as a Case Study

Case study

Khalid Adam
University Malaysia Pahang,
College of Engineering, Department of Electrical Engineering
26300, Pahang, Malaysia
khalidwsn15@gmail.com

Izzeldin I. Mohd
University Malaysia Pahang,
College of Engineering, Department of Electrical Engineering
26300, Pahang, Malaysia
izzeldin@ump.edu.my

Younis Ibrahim
College of IoT Engineering,
Hohai University
Changzhou, Jiangsu 213022, China
Younis@hhu.edu.cn

Abstract *– There have been an extensive use of Convolutional Neural Networks (CNNs) in healthcare applications. Presently, GPUs are the most prominent and dominated DNN accelerators to increase the execution speed of CNN algorithms to improve their performance as well as the Latency. However, GPUs are prone to soft errors. These errors can impact the behaviors of the GPU dramatically. Thus, the generated fault may corrupt data values or logic operations and cause errors, such as Silent Data Corruption. unfortunately, soft errors propagate from the physical level (microarchitecture) to the application level (CNN model). This paper analyzes the reliability of the AlexNet model based on two metrics: (1) critical kernel vulnerability (CKV) used to identify the malfunction and light- malfunction errors in each kernel, and (2) critical layer vulnerability (CLV) used to track the malfunction and light-malfunction errors through layers. To achieve this, we injected the AlexNet which was popularly used in healthcare applications on NVIDIA's GPU, using the SASSIFI fault injector as the major evaluator tool. The experiments demonstrate through the average error percentage that caused malfunction of the models has been reduced from* **3.7***% to* **0.383***% by hardening only the vulnerable part with the overhead only 0.2923%. This is a high improvement in the model reliability for healthcare applications.*

Keywords *– Convolutional neural networks, Reliability, Soft errors, GPUs, healthcare applications*

1. INTRODUCTION

Convolutional Neural Networks (CNNs) are special type DNNs that have shown state-of-the-art results on many competitive benchmarks such as medical image classification [1], pathological brain detection [2], and disease detection [3] among many others. In fact, reports have revealed that CNNs is considered to be more effective, for they own the paradigms of more biologically inspired structures than other traditional networks [4]. This has led to the development of different CNNs including AlexNet [5], VGG [6], and DenseNet [7]. These CNNs derive their competence from being trained to a large database named ImageNet, Large-Scale Visual Recognition Challenge [8]. Hence, they occupy high positions of suitability in modern image classification. In fact, these machine learning networks have the ability to understand hierarchically classified data from lower to higher levels by developing a deep pattern of the input data. Based on these salient features and performance of CNNs, several researchers have exploited them to perform new tasks like the classification of medical images. Specifically, the knowledge acquired when these networks have been trained on millions of images are transferred into new tasks, thereby taking advantage of certain weights of their learned parameters (i.e., Transfer learning).

CNNs are applied to a wide variety of accelerators (i.e. FPGAs, GPUs, DSPs, etc.) each of which has its own elements, behaviour and execution flow. However, due to their computational power, graphics processing units (GPUs) are extensively used in CNNs to overcome the inherent computational challenges of healthcare applications [9] [10] [11] [12]. Notwithstanding, there are certain GPU units that if exposed to soft errors, can disrupt the reliability of the GPU operations; these units include memory elements such as register file and logic resources such as Arithmetic Logic Units (ALUs) [13]. Hence, when using GPUs to accelerate CNNs models in healthcare applications it is important to ensure that potential data corruption is avoided and failure rates must be reduced to the minimum and should not exceed 10 Failures in Time (FIT), which is defined as errors in10^9 hours of operations [14]. Thus, soft errors that occur in GPUs can eventually lead to misclassification of objects in CNNs, and the consequences would be disastrous. For instance, in [15] the authors have reported instances of the da Vinci robotic surgical system adverse events that included some kind of patient injuries and death, and reported as a "Malfunction", "unanticipated" and "unintended" errors. Food and Drug Administration (FDA), reported that 1078 of the adverse events (10.1%) were unintended errors (soft errors) happened, including 52 injuries and 2 deaths [16].

In this contribution, the reliability of the AlexNet model on a GPU was anayzed by conducting series of fault-injection campaigns, using NVIDIA's SASSIFI The first significant contribution of this study is the determination of soft errot resilience in AlexNet through comprehensive analysis, and ranking of vulnerable model parts from the perspective of kernel and layers. The second contribution is reduction of soft errors through a Selective Hardening approach. The subsequent sections of this paper are includes Section 2 related work. Section 3 which is brief background of AlexNet, Graphics Processing Units, and Soft Errors Propagation in GPUs. Section 4 desribes the Selective Hardening Strategy, and Section 5 contains the methodology while Section 6 presents the results generated from analysis of the Kernels and Layers. The experimental results and their analysis is presented in Section 7, whereas the Time Overhead Execution comparison is presented in Section 8 and the conclusion is in Section 9.

2. RELATED WORK

There are several studies in [17][18][19] [20][21] authors evaluated and analyzed the reliability of CNN models. Hence, it has been established that there are varieties of CNN architectures, with each having different bevavior and workflows. The different CNNs have been implemented on various accelerators including GPUs, ASICs, and TPU, through their peculiar execution flow based on their varying components. This makes it difficult to directly generalize the case of a particular CNN to other architectures [22]. Andru et al. [23] proposed a CNN architecture called EndoNet to automatically recognize the presence of surgical tools. The model trained on the Cholec80 dataset. But, the authors address the reliability of the proposed model in terms of temporal precision, which is different from our perspective. Amy et al. [24] introduce an approach to analyzing and tracking the movements of the surgical tool. They used the CNN model and m2cai16 dataset to train the model. However, the authors did not consider the reliability of the model. Grewal et al. [25] described an approach for automated brain hemorrhage detection from computed tomography. The study used DenseNet201 architecture as an emergency diagnosis tool, but the authors did not consider the reliability of such a model for the intended application, which is actually a safety-critical application, based on real-time CNN model detection.

In another study by Dunnmon et al. [26], three CNN models (Alexnet, ResNet, and DenseNet201) were used to classify chest radiographs into groups categorized as either normal or abnormal. This approach can help to prioritize abnormal chest radiographs automatically. In addition, the use of chest radiographs to predict multiclass thoracic diagnosis was reliably addressed in the study. Notwithstanding, the reliability of the models has not been considered. In another study, robot-assisted surgery was proposed by Wang and Fey [27]. Specifically, a deep analytical framework for learning and assessment of skills in surgical training was implemented. The individual skill levels in multivariate data with various time series were mapped to the motion kinematics using a deep CNN. Interestingly, instantaneous feedback can be obtained from personalized surgical training if the model is incorporated into the robot-assisted surgical systems pipeline. However, the reliability of the model to the intended application was not considered by the authors. Notably, several approaches have been developed to reduce the occurrence of the soft error in GPUs, through software solutions. This include Double Modular Redundancy (DMR) and Triple Modular Redundancy (TMR). However, the major drawback to the use of these solutions is the runtime overheads.

3. BACKGROUND

3.1. ALEXNET NEURAL NETWORK

AlexNet is a CNNs architecture, composed of eight layers with weights; the first five are convolutional layers and the remaining three are fully connected layers. The fully-connected layers generally consist of 4096-dimensional features [5]. AlexNet has been confirmed to be suitable for classifying medical images for diseases like lung diseases, heart challenges and, cancer. As presented in (Fig. 1), the input image in AlexNet should be augmented to an image size of 227 x 227 x 3. The window shape size applied to the first layer 96 convolution filter is 11 × 11, whereas it is 5×5 in the second

layer 256 convolution filter. Subsequently, 3×3 window size is applied to the remaining 384, 384, and 256 convolution filters, respectively. A maximum pooling layer of 3×3, with 2 strides is present in the network after the first, second, and last convolutional layers. Asides these five convolution layers, 4096 neuron outputs are present in the seven fully-connected layers following the fifth convolutional layer. Then, one fully connected output layer is situated at the end of the network which initially contains 7 output classes. Generally, an excellent performance of tasks involving computer visions can be achieved by using important keys like Dropout, ReLU, and preprocessing. The pre-trained AlexNet model with and weight configuration can be found on the Darknet framework.

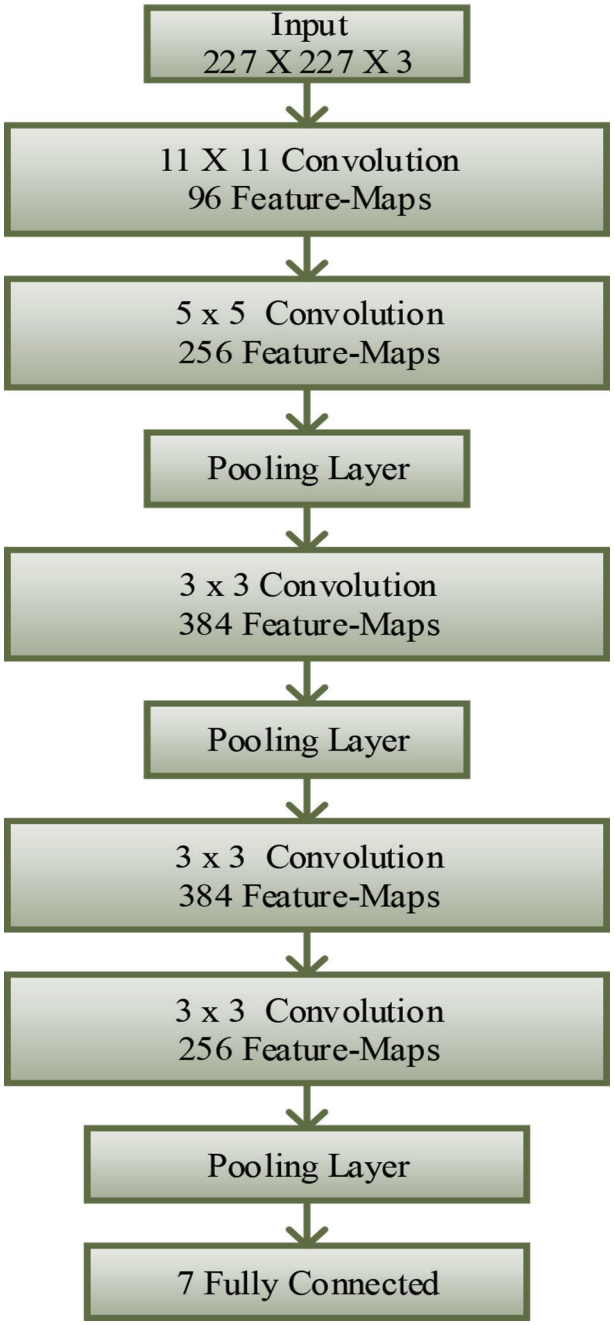


Fig. 1. AlexNet Architecture

3.2. GRAPHICS PROCESSING UNITS

The use of GPUs have recently extended beyond being used solely for graphics tasks to being used in more general-purpose devices. In fact, currently considered as the main DNN accelerators [28]. The increased interest in GPUs is mainly associated with its great computational power and massively parallel architecture. Based on these, they are more preferable in algorithms that require intense computing such as neural networks. It can be seen in (Fig. 2) that the basic GPUs architecture is based on generation of Compute Unified Device Architecture (CUDA) as the programming model. The basic building block unit in GPUs is known as Streaming Multiprocessor (SM). The SM is made up of several components such as streaming processors (SPs) that are used in arithmetic calculations, and special function units (SFUs) which function for sine, cosine, and square root operations. In addition, the SM comprises load/store (LD/ST) units used in memory operations, as well as many other registers for caches. SMs are the core idea of parallelism in GPUs. The basic structure of SMs in the GPU architecture is shown in Fig 2, each of which can only execute one thread in a clock cycle with dedicated registers from the register file. The warp scheduler to be executed next on the CUDA cores in a given SM selects a group of 32 threads (called a warp), and then instructions are dispatched by the instruction dispatch unit. The threads in each warp execute in a SIMT (single instruction, multiple threads) fashions. The global system memory of a GPU is located in the dynamic random-access memory and the global memory would normally be accessible to the SM. The L2 cache is a shared memory mainly shared by the SPs in the SMs.

Hence, read and write instructions can be executed at the L2 cache level by each SM. On the other hand, all the SPs in an SM can access the register file. The register file is basically mapped in the SMs to enhance computational performance through data caching for the running threads on each SM.

3.3. SOFT ERRORS PROPAGATION IN GPUS

The features of modern GPUs can be significantly affected by radiation strikes either in space or on earth and this can invariably result in computational failure or data corruption. Therefore, one of the notable unreliability sources in modern systems is soft errors. This is because electronic devices like GPUs would malfunction when they are struck by high-energy particles [29]. Usually, the tolerance to failure in safety-critical systems is restricted to 10 Failures in Time (FIT). Therefore, since soft errors in DNN accelerators (GPU inclusive) are more deleterious compared to other electronic devices, there is a need to pay deliberate attention to them for two main reasons.

Firstly, the complex memory hierarchy is used for improved latency. Secondly, the massively-parallel structure of the GPUs, that tends to disperse a single fault

to multiple faults. Usually, when memory elements of a GPU (indicated by the red sections in Fig. 3) are hit by particles, it can significantly affect all the threads that utilize such storage components. When functional components of the GPU such as ALU (INST) or Floating-point (FP) units are hit by a particle, it creates temporary-voltage pulses, Single Event Transients (SETs). The SET can then travel through the logic components of the GPU where it can be captured by a storage component. Specifically, a latch or flip-flop would trigger the flip of one or more bits from one value to another such as from 0 to 1 or from 1 to 0. However, this can be curtailed by applying a fault detection technique such as DMR, TMR, and ABFT. Generally, faults generated in GPUs in form of data values or logic operations often results in errors like Silent Data Corruption (SDC), system hang otherwise called Detected Unrecoverable Error (DUE), or outright crash of the application.

However, the errors might sometimes not result in an observable error in which case it is known as Masked errors. The propagation of errors could be via different processes, in this case, layers, until they arrive at the program output (AlexNet) where they will eventually lead to problems such as the misclassification of objects. Due to this, soft error in GPU is a very critical issue in safety-critical healthcare applications where high reliability is generally required. This is mainly due to the fact that even small errors might lead to serious injury or death as reported by the Food and Drug Administration (FDA) department [30].

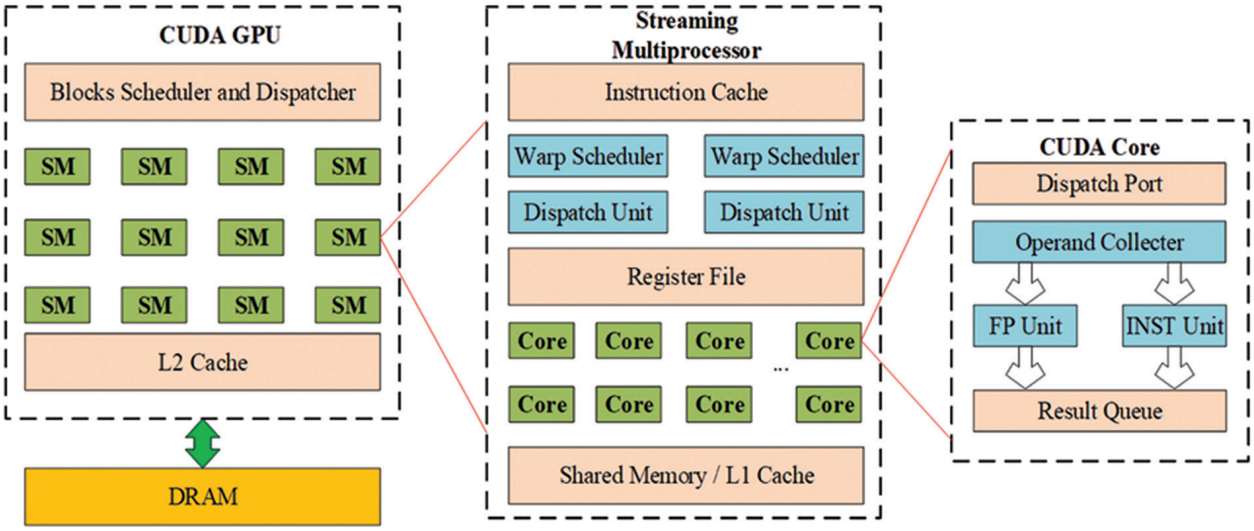
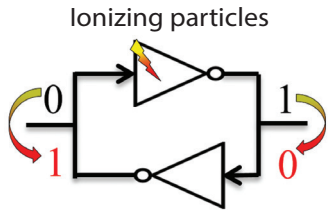


Fig. 2. GPU architecture and memory



(a) Single Memory cell (bit)

1	0	0	1	1	0	0	1
1	0	0	0	1	0	0	1

(b) Memory Byte (8-bits)

Fig. 3. Memory-element strikes

4. SELECTIVE HARDENING STRATEGY

In this section, we developed Selective Hardening Strategy (SHS) by identifying the most vulnerable kernels for AlexNet, via fault injection (soft errors). In order to identify the most vulnerable kernels in the AlexNet model, we present a methodology (section 4). The key concept of SHS is that it is based on the well-known Triple Modular Redundancy (TMR), and it intertwines three copies of the instructions and adds majority voting. In short, this SHS mitigation consists of triple kernels, by means of majority voters. Based on this concept, our strategy is a selective solution that protects only the vulnerable kernels instead of duplicating the whole as in DMR and TMR, to reduce the overheads, and thereby offering more flexibility to designers.

5. METHODOLOGY

5.1. DATASET

The AlexNet model was trained and evaluated with a Cholec80 dataset. Eighty cholecystectomy videos which were recorded at the Strasbourg University Hospital (25 fps) are contained in the dataset [23]. In addition, tool presence annotation is present in the dataset at 1 fps. Seven different tools were utilized in the dataset including hook, clipper, irrigator, bipolar, grasper, specimen bag, and scissors. If at least half the tip of the tool appears in the scene, then it is considered a present tool. The AlexNet was trained with the first forty videos while the remaining videos were used for validation.

5.2. FAULT-INJECTION SETUP

A Maxwell architecture-based GPU GTX 750 Ti was used for this purpose [31], with a SASSIFI fault injector which was primarily used to assess the reliability of AlexNet model runs on GPUs. This was achieved through fault injection campaigns which made it easier to determine the possibility of a low-level corruption to propagate to the output. Notably, the AlexNet model was trained on the Cholec80 dataset for the surgical tool detection. With this tool, it was possible to carry out fault injection through three distinct modes. The first mode is the Register File (RF) mode which was used to determine the Architectural Vulnerability Factor (AVF) of the register file, as well as the response of our model (AlexNet), to errors present in the memory elements. The second mode is the Instruction Output Address (IOA) mode while the third mode is the Instruction Output Value (IOV) mode. The second and third modes (IOA and IOV) were used evaluate the Program Vulnerability Factor (PVF). In addition, they were used to investigate how a single error modifies the result of instruction and propagates to the program output (AlexNet).

A total of 1000 faults was injected for each of the three modes RF, IOA, and IOV. this number of injections was enough to guarantee that the worst-case statistical error bars at 95% confidence are at 1.96%. Notably, various bit-flip models can be obtained from SASSIFI including zero value, single bit-flip, a random value, and multiple bit-flip. Nevertheless, only the single bit-flip and random value models were selected for the three injection modes of this present study. These models were preferentially selected because single-bit flip is more suitable and realistic for memory errors. On the other hand, all the other three bit-flip models are represented by the random value. The AVF of the register file is measured when errors are injected with RF mode while the PVF of the algorithm is measured if the errors are injected with IOA and IOV.

As a consequence of fault injection and comparison of program output with the golden output (i.e., the pure outcome), three categories are expected, Masked, DUE, or SDC. It should be noted that SDC is the only error of interest to this study when studding the error propagation within the model. This is because crash and hand errors (DUEs) are not being propagated to a subsequent layer. Similarly, masked errors are instantaneously masked at the location of occurrence. The SDC errors and the mechanism of their propagation through layers were further grouped into three categories, for a better understanding of the concept. The first group is the Malfunction SDCs which represent errors that propagate, arrive at the program output, and alter the probabilities vector thereby impacting the object's rank via misclassification. The second groups are the Light-Malfunction SDCs. These are errors that propagate, arrive at the program output, and alters the probabilities without changing the object's rank. This situation is otherwise called tolerable SDC meaning object misclassification did not occur. The third groups are the No-Malfunction SDCs which are the error that propagates without reaching the final program output. This means the errors are masked in some layers but this group of SDCs are different from Masked errors which do not propagate at all.

6. KERNELS AND LAYERS FAULT INJECTION RESULTS AND ANALYSIS

The results generated from our fault-injection campaign, and their analysis are presented in this section. A detailed sensitivity analysis was conducted to assess the resilience of the AlexNet model. This was performed through the evaluation of two metrics. The first metric is the critical kernel vulnerability (CKV) which helps to recognize the presence of malfunction and light-malfunction errors in each kernel. The second metric is the critical layer vulnerability (CLV) which enables tracking of malfunction and light-malfunction errors within the layers.

6.1. CKV

As discussed in section 2.1, several layers are present in the Alexnet model. Implementation of these layers on a GPU would result in the generation of several kernels (special functions) for each layer. Notably, the kernel is a component of the source code that is implemented on a GPU, not a CPU and the kernels have distinctly peculiar computing characteristics. Therefore, all the static kernels required for a particular task is needed for an in-depth analysis of the vulnerability levels of the different kernels to malfunction and light-malfunction errors. This is also required to determine how they influence the final output of the model through object classification. Nevertheless, we only consider the kernels that are required for inference whereas the kernels used for training not incorporated as presented in Table 1. After fault injection, each kernel of the injected program is analyzed and the most vulnerable kernels of our AlexNet are determined. It is worth noting that in Fig. 4, Fig. 5, and Fig. 6, the probabilities of the whole graph sum up to 100% rather than the vertical bars of each kernel. This is because the AlexNet model program consists of all these kernels in Table 1. In other words, AlexNet programs are divided into small pieces of programs (kernels) that are executed in the GPU.

The kernels that produce a larger amount of errors can be easily identified by observing Fig. 4, Fig. 5, and Fig. 6. Likewise, the resilient kernels can be seen in the stated figures. Generally, Malfunction, Light-Malfunction, and DUEs in RF, IOA, and IOV are noticeable for all the kernels. However, the two kernels with the highest vulnerability in the AlexNet model are Im2col and Add_bias. In contrast, only a small number of DUEs, Malfunction and Light-Malfunction are noticeable in the other kernels which indicates that they are highly resilient to soft errors. Similarly, the Fill kernel

shows little small Malfunction and Light-Malfunction error which suggests that they are highly resilient to soft errors. It can also be observed that the building of one CNN layer such as Conv. Or activations layers can be obtained from the contributions of more than one kernel. Statistically, it is easier to identify the layers that are more susceptible to faults, thereby facilitating decision-making at the error mitigation step. It is evident in Fig. 4 that the kernels in the RF mode produce larger number of DUE errors, compared to Malfunction and Light-Malfunction errors. In contrast, Fig. 5 and Fig. 6 shows that the kernels in IOA and IOV modes tend to produce higher levels of Malfunction and Light-Malfunction errors compared to DUE errors. The reason for this behavior as reported by [32] is that RF injection is the lowest injection level whereas higher injection levels are performed at IOA and IOV sites because instructions are manipulated. Generally, the result discussed here indicates that different vulnerabilities are associated with the static kernels of the AlexNet model. These results make it easy to determine the kernels that need to be duplicated as a means of saving costs, rather than duplicating the entire, which is a very costly process.

Table 1. AlexNet inference kernels and their corresponding layers

Layer	Kernel	Kernel Task
Convolutional	Im2col_gpu, Add_bias, Fill_gpu	Operation to matrix-multiplication operation and add biases to the necessary parameters after the matrix multiplication
Max pooling	Forward_ maxpool	To reduce the spatial dimension of the input volume for next layers
Activation	Activation_array	Apply nonlinearity to the feature maps to reduce the input linearity for the next layer
Softmax	Softmax	To calculating the probabilities of each class

6.2. CLV

For a detailed understanding of the mechanism of error propagation through the layers of AlexNet to the output, the PVF was measured while the fault-injection mode is in the IOA and IOV. Fig. 7 to Fig. 9 presents the error propagation at each AlexNet layer, showing their different sensitivity to soft errors. Hence, the errors (Malfunction and No-Malfunction) that are propagated from the injected layers can be tracked to the last layer. The three SDC categories were investigated for each layer and the categories were calculated as discussed in section 4.1. The AVF and PVF values were measured for the given errors to determine the layers with a high probability to generate errors that can significantly alter the model prediction (object misclassification). The first observation is that the probability values of each layer do not sum up to 100%. The reason is that if a layer is injected, it produces three types of errors: DUE, Masked, and SDC. However, because in this subsection, we intend to analyze the propagation of errors, and only SDC errors propagate through layers. Therefore, we shall mainly focus on the three SDC types which are Malfunction SDCs, Light-Malfunction SDCs, and No-Malfunction SDCs. This is the reason why they do not sum up to 100%.

It is evident in Fig. 7 that in RF mode, the layers only tend to generate zero amounts of Malfunction. Layers 0 to 6 generates big amounts of Light-Malfunction on an average 12.57% AVF. On the other hand, the layers with No-Malfunction is on the average of 3.9%. This is an indication that RF injections present a less significant influence on the resilience of layers against Malfunction errors. However, different amount of DUE errors is produced by the layers, due to reasons earlier stated in Section 6.1. The injection of faults into the IOA and IOV generates SDC errors in different layers, and this influences their resilience. Therefore, these two modes are discussed and analyzed in further detail. However, a lesser percentage of DUE errors was generated by the IOA and IOV injections, compared to RF mode.

As can be seen in Figure 8, in the IOA mode, layers 0, 1, and 7 generate Malfunction errors of 0.2% and this value is considered too high in safety critical applications. However, they still generate Light-Malfunction with an approximate value of 18.6% due to the AlexNet structure. On the other hand, 67.9% of the injected faults represent No-Malfunction. Likewise, several percentages of DUE errors are produced by the layers on the average 1%. However, about half of the layers including layer 7, 8, 9, 10, 11, 12, and 13 are not significantly affected by DUE errors. As illustrated in Fig. 9 for the IOV mode, an average Malfunction value of 0.8% was generated by the layers. This significantly impacts the reliability of the model which suggests that the percentage is unacceptable. About 3.4% of the total Malfunction generated by the model was statistically contributed by layers 0 and 2. In addition, the largest percentage of Light-Malfunction and No-Malfunction were produced by these two layers which indicates that they are more vulnerable than other layers of the model. In SASSIFI, IOV mode is considered to be the highest injection level among the three modes and this is why layers injections performed through IOV mode is most unlikely to terminate the model execution. Generally, from the IOA and IOV model results presented in Fig. 8 and Fig. 9, layers 0 and 2 generate more Light-Malfunction errors and No-Malfunction compared to

other layers present in the network. In addition, the convolutional (11x11 and 5x5) of the AlexNet as described in Fig. 1 is represented by these layers. Based on the explanation in Section 2.1, linear function is the activation function within these layers, and this equates the output and input thereby retaining the initial size. This observation clearly describes the structure of the AlexNet model itself, whereby larger input sizes are possessed by these layers. The direct relationship between the execution time of each layer and the corresponding exposure time in soft errors is not surprising because longer layer exposure should expectedly produce higher rates of Malfunction errors. It is noticeable that relatively fewer errors are generated at the layers close to the output layer. This is probably due to the gradual reduction of the 227×227×3 matrix input size at the first layer as it propagates and reaches the output layer where the size is 13x13x256, before becoming a vector of 7 probabilities.

Most of the SDC errors (all categories) at the AlexNet output originate from faults that are related to the Conv. layers. Considering the size of input and filter numbers, it is evident that all the convolutional layers utilize the same kernels and possess the same AVF and PVF values. Nevertheless, Fig. 7, 8, and 9 shows that the possibility of an output model being impacted by injection is mainly dependent on the position of the layer in the network. This is particularly evident in layer 7 which is located just after convolutional layers. Herein, Softmax is observed to be the most reliable layer because no error is generated in this layer and most of the Malfunction and Light-Malfunction are masked. This can be attributed to two main reasons. Firstly, contrary to the case for other layers, Softmax is invoked just once and this significantly reduces its execution time thereby making it almost impossible for errors to generate in it. Secondly, the vector probability functionality of Softmax sums up to one. Therefore, even if SDC (Malfunction and Light-Malfunction) alters the input value of Softmax, there could still be retention of a probability percentage such that error will be No-Malfunction. The scores of a vast majority of the output-probability are zeros because the classified object should bear similarity with some of the remaining objects among the 7 classes needed for classification. This is a confirmation of the underlying feature of the ANNs, which is fault tolerance. This means that if the values are negative, they represent No-Malfunction.

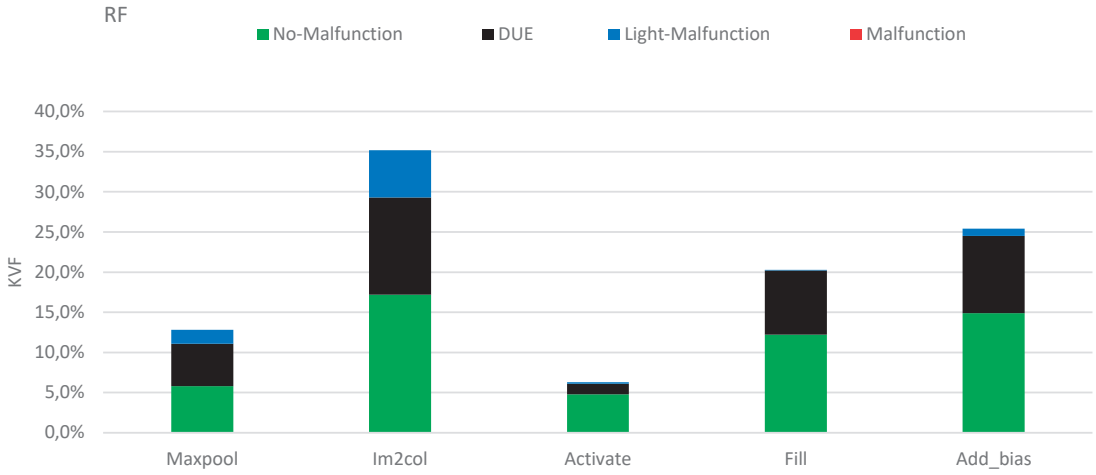


Fig. 4. Kernels vulnerability of AlexNet models for RF mode

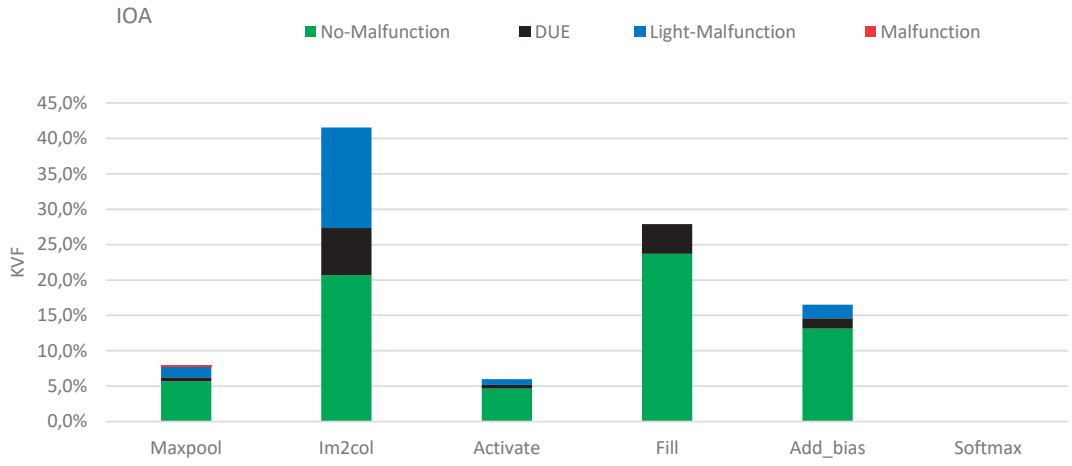


Fig. 5. Kernels vulnerability of AlexNet models for IOA mode

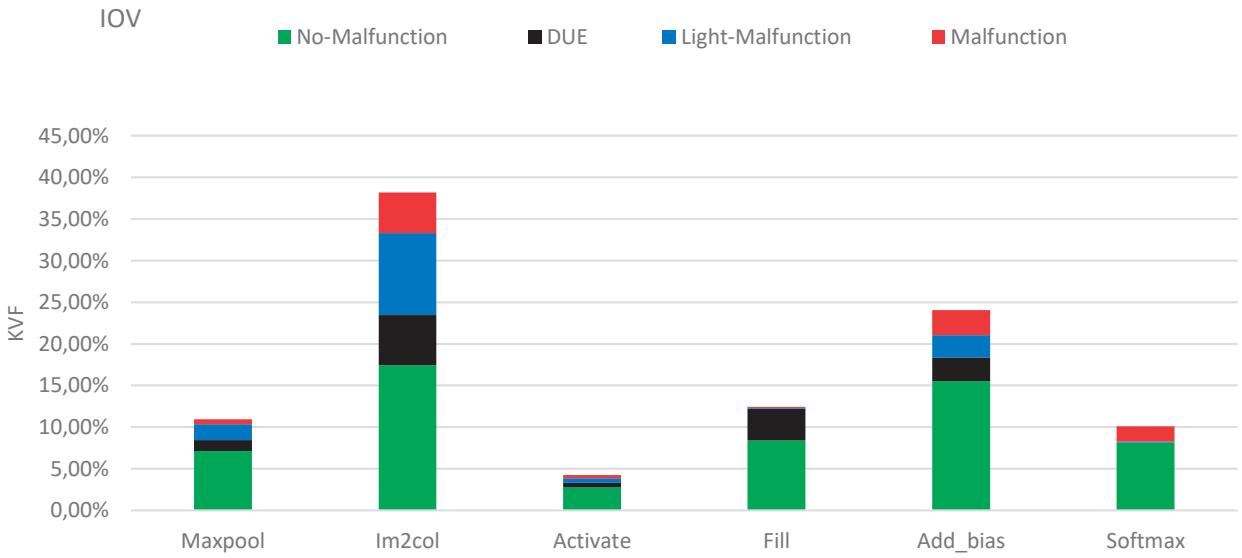


Fig. 6. Kernels vulnerability of AlexNet models for IOV mode

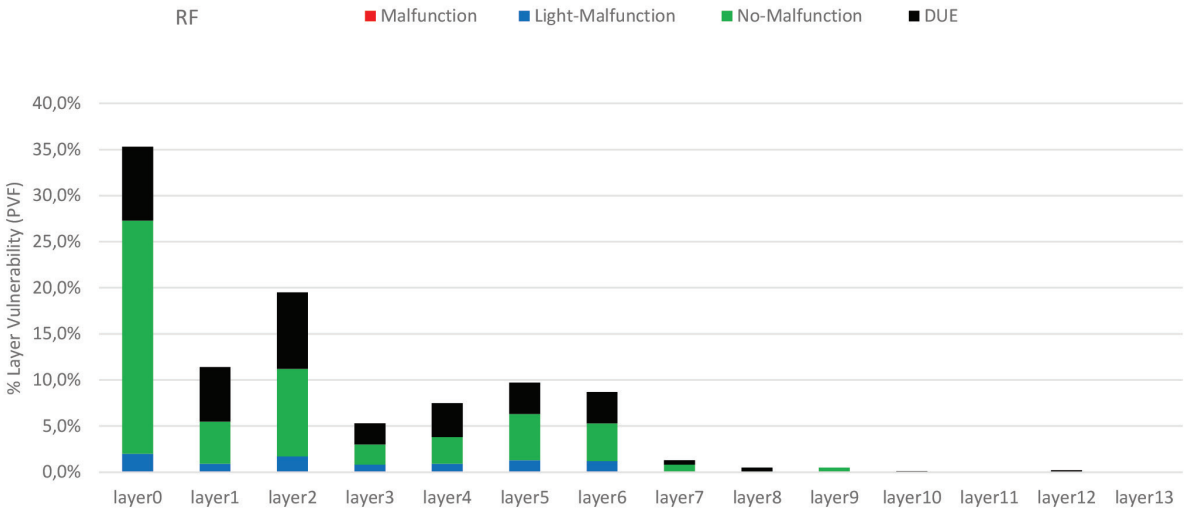


Fig. 7. AVF of RF Mode layer for Malfunction SDCs,
Light-Malfunction SDCs, No-Malfunction SDCs and DUE

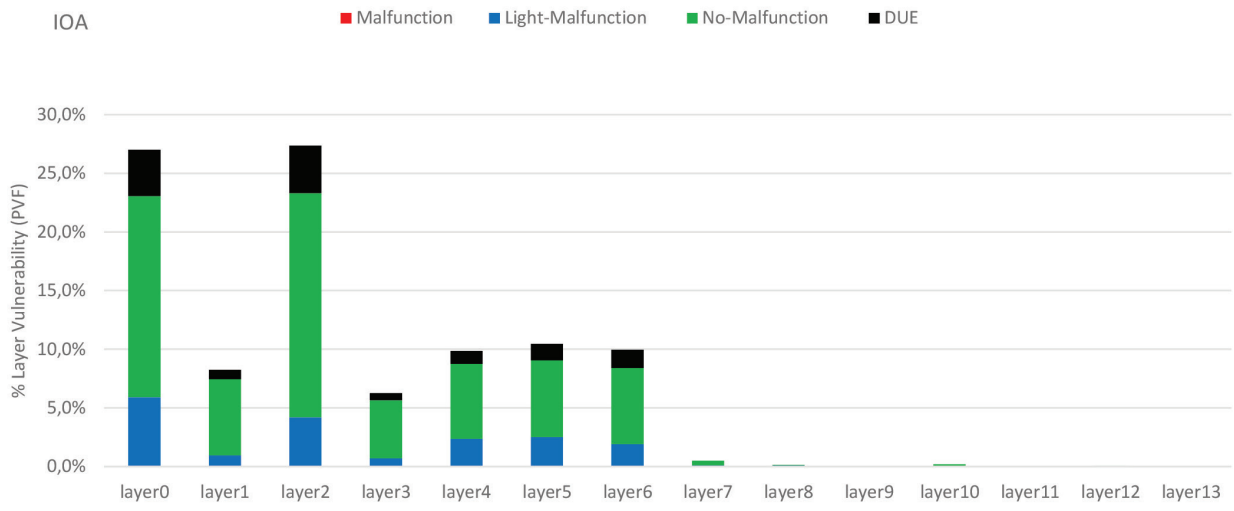


Fig. 8. PVF of IOA Mode layer for Malfunction SDCs,
Light-Malfunction SDCs, No-Malfunction SDCs and DUE AlexNet

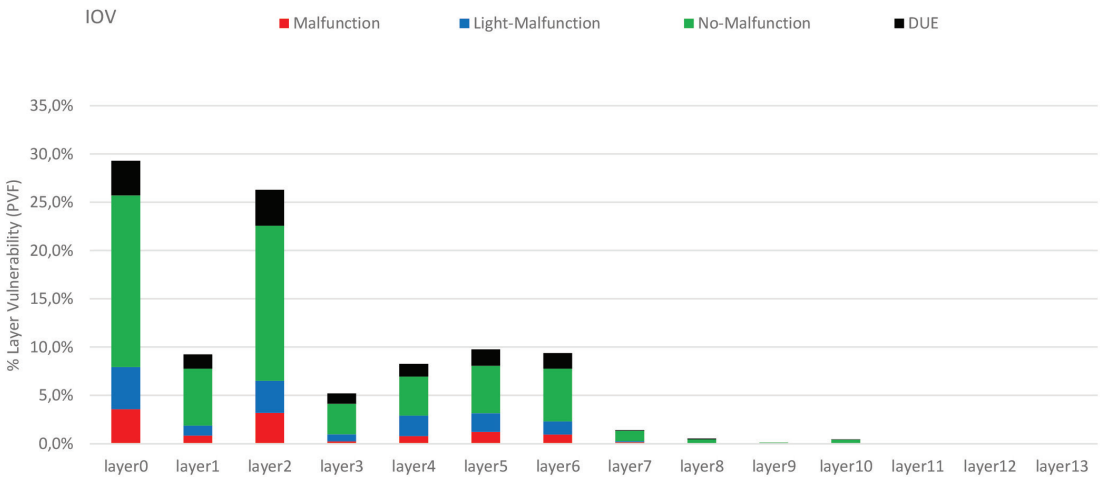


Fig. 9. PVF of IOV Mode layer for Malfunction SDCs, Light-Malfunction SDCs, No-Malfunction SDCs and DUE

7. EXPERIMENTAL RESULTS AND ANALYSIS WITH SHS

7.1. CKV ANALYSIS

In RF, IOA, and IOV kernels in Fig. 10, Fig. 11, and Figure 12 respectively, we evaluated the kernels by applying our SHS. Based on our analysis in section (5.1) the top-2 vulnerable kernels for AlexNet are Im2col and Add_bias in all three modes. All the Malfunction SDCs in these kernels become No-Malfunction. Our technique shows significant improvement in the AlexNet model, the errors in top-2 vulnerable kernels (Im2col and Add_bias). The errors in RF mode reduce from 5.90% to 0.00% Light-Malfunction in Im2col and from 0.90% to 0.00% Light-Malfunction in Add_bias, while there is not any modification on both kernels in Malfunction errors (still zero). Whereas, the errors in IOA mode reduce from 14.20% to 0.00% Im2col and 1.90% to 0.10% Add_bias in Light-Malfunction. Meanwhile, there is not any change in both kernels in Malfunction errors (still zero). They are also significantly improved in IOV mode, where errors are reduced from 9.82% to 0.78% in Im2col and 2.69% to 0.12 in Add bias in Light-Malfunction.

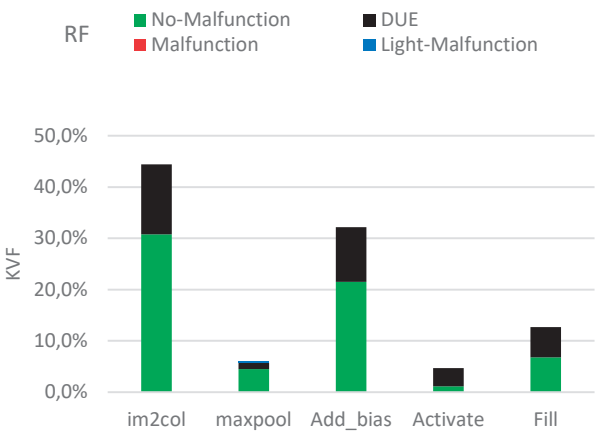


Fig. 10. Kernels vulnerability of AlexNet models for RF mode after applying our SHS

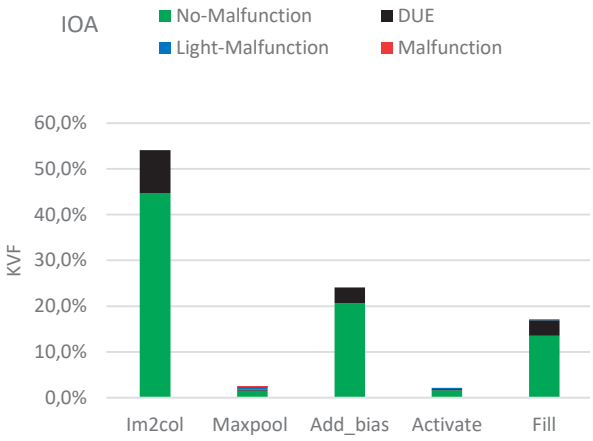


Fig. 11. Kernels vulnerability of AlexNet models for IOA mode after applying our SHS

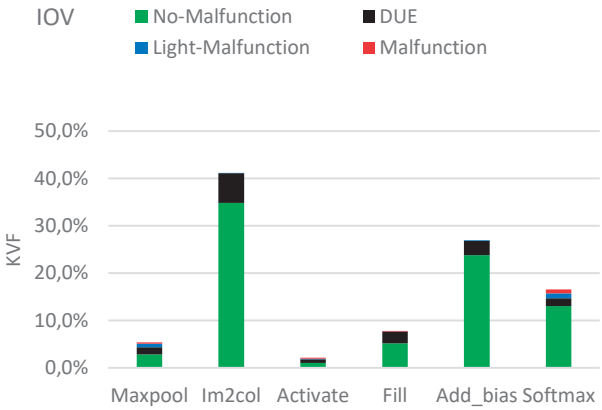


Fig. 12. Kernels vulnerability of AlexNet models for IOV model after applying our SHS

7.2 CLV ANALYSIS

In this subsection, the resilience of the AlexNet was reevaluated from a layer perspective through analysis of the resilience of the layer after applying our technique. As error propagates through layers is the main target in this phase, we measure all types of SDC errors including

Malfunction SDCs, Light-Malfunction SDCs, No-Malfunction SDCs, and DUEs. Fig. 13 shows the RF model after applying our mitigation technique, the experimental result shows only 0.00% and 0.30% errors of Malfunction and Light-Malfunction respectively. While it still produced a big amount of No-Malfunction SDCs in the percentage of 62.80 %. Despite the amount of the DUEs still high at 36.90%. On the anther hand, Fig. 14 and Fig. 15 show IOA and IOV, both of the modes produced less amount of the DUEs 16.55% and 15.6% respectively, compared to the RF mode, and the reason that IOA and IOV have a different level of injections. Therefore, in Fig. 14 IOA, produced Malfunction and Light-Malfunction on average 0.01% and 0.09% respectively, meanwhile most of the errors 82.10% No-Malfunction SDCs. On the anther hand, IOV in Fig. 15 produced less amount of the Malfunction and Light-Malfunction on average 0.08% and 0.19% respectively, where 80.76% of the errors No-Malfunction.

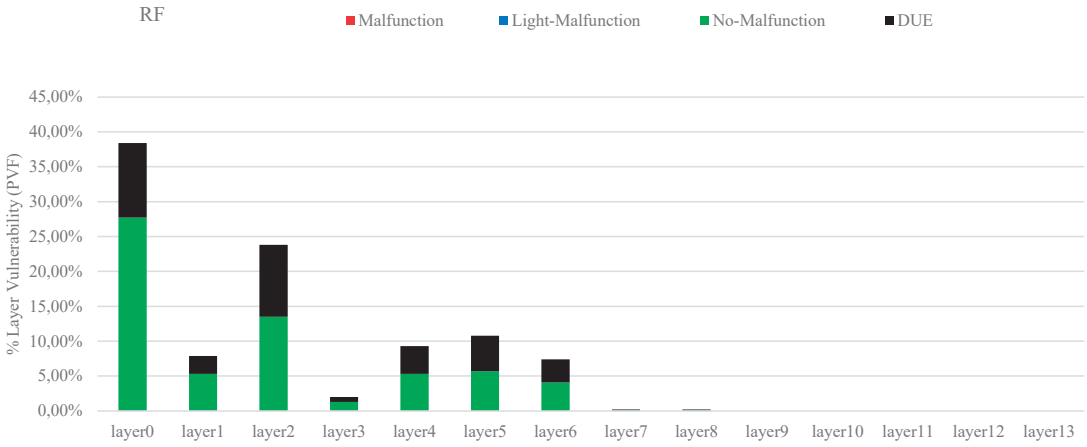


Fig. 13. AVF of each layer (after applying our SHS)

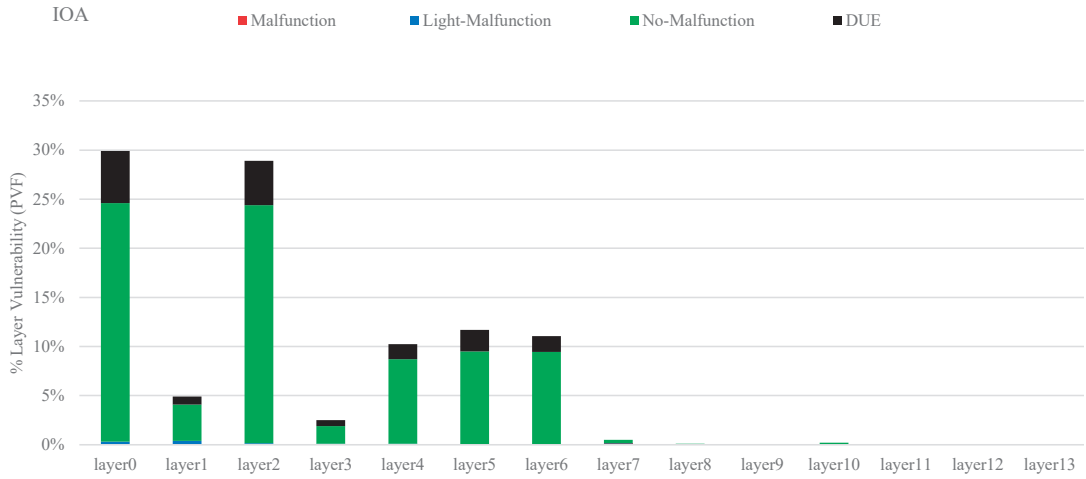


Fig. 14. PVF of each layer (after applying our SHS)

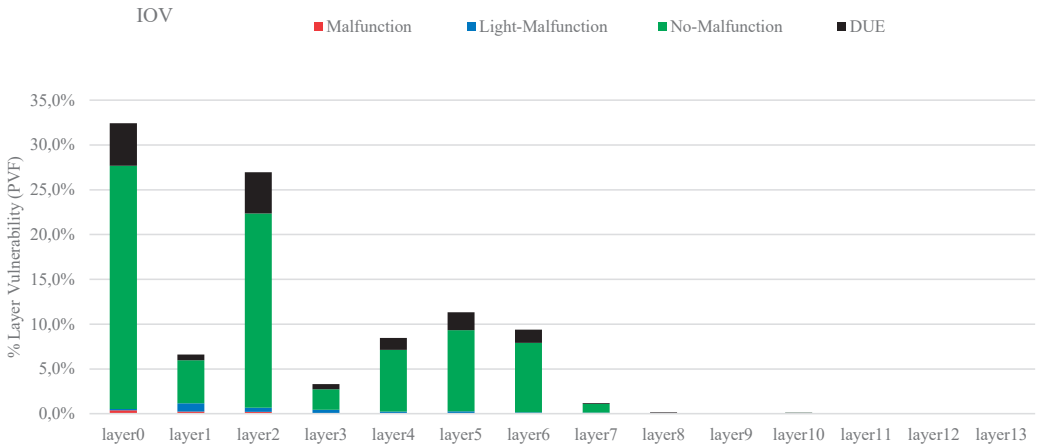


Fig. 15. PVF of each (after applying our SHS)

In this section, we evaluate our proposed solution by measured the performance overheads for whole models (AlexNet) and vulnerability kernels. By calculating the execution time (performance) for the whole model and the vulnerable kernels before and after implementing our technique. Table 2 shows the overheads of our technique, DMR, and TMR techniques summarizes the error injection, and each kernel after applying our mitigation strategy technique. As our technique to achieve a low-overhead with sufficient reliability, we selectively hardened only the vulnerability kernels. Consequently, the overhead can be reduced, by exploiting the SHS that is executed and overhead has only increased by 0.2923%, thus improved the reliability of models. Compared to the DMR and TMR whereas the overhead increased 97.461% and 200.881% respectively. Therefore, our technique shows highly significant error mitigation by only hardened selective kernels. And removed the unnecessary overhead especially in the safety-critical system (healthcare applications) that comes with strict deadlines, the overhead associated with duplication whole model is unacceptable.

Table 2. Comparison of overhead of Unhardened model, S-MTTM-R, DMR and TMR

Kernels (Time by MS)	Number of invocations	Unhardened	SHS	DMR	TMR
Im2col	5	0.0002303	0.0104156	0.0208312	0.0312468
Add_bias	5	0.000708	0.0035207	0.0070414	0.0105621
Maxpool	4	0.0237066	0.0237066	0.0474132	0.0711198
Activation	8	0.4267208	0.3935817	0.7871634	1.1807451
Fill	5	0.0948266	0.14224	0.2844858	0.4267258
Softmax	1	0.1201457	0.0948267	0.1896534	0.2844801
The whole model	1	0.666338	0.6682913	1.3157572	2.0048797

8. CONCLUSION

In this contribution, we analyzed and evaluated the Malfunction and Light-Malfunction SDCs of soft errors for the AlexNet model on the GPU. Our CKV and CLV identify the most vulnerable kernels and layers. Based on the analysis in Section 5 on the reliability of the model's bit sensitivity, the vulnerable bits can be selectively protected using SHS. Our result shows a high reduction rate of errors in the top vulnerable kernels such as Im2col and Add_bias. Besides, the model achieved a high reduction in No-Malfunction from 54.9%, 67.9%, and 59.4% to 62.80%, 82.10%, and 80.76% in the three modes such as RF, IOA, and IOV, respectively. Moreover, the performance overhead of our solution is compared with the well-known protection techniques such as Algorithm-Based Fault Tolerance (ABFT), Double Modular Redundancy (DMR), and Triple Modular Redundancy (TMR). The proposed solution shows the least overhead while correcting up to about 82.8% of the SDC errors in a CNN, thereby remarkably improving the healthcare domain's model reliability.

9. ACKNOWLEDGEMENT

This research was supported by Universiti Malaysia Pahang, through the UMP internal grant (PGRS190325)

10. REFERENCES

[1] R. M. Haralick, I. Dinstein, K. Shanmugam, "Textural Features for Image Classification", IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-3, No. 6, 1973, pp. 610–621.

[2] S. Lu, Z. Lu, Y. Zhang, "Pathological brain detection based on AlexNet and transfer learning", Journal of Computational Science, Vol. 30, 2019, pp. 41–47.

[3] F. Demir, A. Şengür, V. Bajaj, K. Polat, "Towards the classification of heart sounds based on convolutional deep neural network"", Health Information Science and Systems, Vol. 7, No. 1, 2019, pp. 1–9.

[4] W. Rawat, Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review", Neural computation, Vol. 29, No. 9, 2017, pp. 2352–2449.

[5] A. Krizhevsky, I. Sutskever, G. E. Hinton, "ImageNet classification with deep convolutional neural networks", Communications of the ACM, Vol. 60, No. 6, 2017, pp. 84–90.

[6] A. Vedaldi, A. Zisserman, "Vgg convolutional neural networks practical", Department of Engineering Science, University of Oxford, 2016, p. 66.

[7] G. Huang, Z. Liu, G. Pleiss, L. Van Der Maaten, K. Weinberger, "Convolutional Networks with Dense Connectivity", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019, pp. 1–1. (in Press)

[8] O. Russakovsky et al., "Imagenet large scale visual recognition challenge", International journal of computer vision, Vol. 115, No. 3, 2015, pp. 211–252.

[9] T. Kalaiselvi, P. Sriramakrishnan, K. Somasundaram, "Survey of using GPU CUDA programming

model in medical image analysis", Informatics in Medicine Unlocked, Vol. 9, 2017, pp. 133–144.

[10] A. A. Shvets, A. Rakhlin, A. A. Kalinin, V. I. Iglovikov, "Automatic instrument segmentation in robot-assisted surgery using deep learning", Proceedings of the 17th IEEE International Conference on Machine Learning and Applications, Orlando, FL, USA, 17-20 December 2018, pp. 624–628.

[11] E. Kuznetsov, V. Stegailov, "Porting CUDA-Based Molecular Dynamics Algorithms to AMD ROCm Platform Using HIP Framework: Performance Analysis", Proceedings of the Russian Supercomputing Days, Moscow, Russia, 23-24 September 2019, pp. 121–130.

[12] Tsinghua Univesity, Beijing innovation center for future chips, "White Paper on AI Chip Technologies", 2018. (Online: https://www.080910t.com/downloads/AI%20Chip%202018%20EN.pdf).

[13] D. A. G. De Oliveira, L. L. Pilla, T. Santini, P. Rech, "Evaluation and mitigation of radiation-induced soft errors in graphics processing units", IEEE Transactions on Computers, Vol. 65, No. 3, 2016, pp. 791–804.

[14] D. A. G. Oliveira, S. Member, L. L. Pilla, T. Santini, S. Member, P. Rech, "Evaluation and Mitigation of Radiation-Induced Soft Errors in Graphics Processing Units", Vol. 9340, No. C, 2015, pp. 1–14.

[15] H. Alemzadeh, J. Raman, N. Leveson, Z. Kalbarczyk, R. K. Iyer, "Adverse events in robotic surgery: A retrospective study of 14 years of fda data", PLoS ONE, Vol. 11, No. 4, 2016, pp. 1–20.

[16] H. Alemzadeh, J. Raman, N. Leveson, Z. Kalbarczyk, R. K. Iyer, "Adverse events in robotic surgery: A retrospective study of 14 years of fda data", PLoS ONE, Vol. 11, No. 4, 2016, pp. 1–20, 2016.

[17] F. F. dos Santos et al., "Analyzing and Increasing the Reliability of Convolutional Neural Networks on GPUs", IEEE Transactions on Reliability, Vol. 68, No. 2, 2018, pp. 663–677.

[18] G. Li et al., "Understanding error propagation in Deep Learning Neural Network (DNN) accelerators and applications", Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, Denver, CO, USA, 12-17 November 2017, p. 8-19.

[19] L. Weigel, F. Fernandes, P. Navaux, P. Rech, "Kernel vulnerability factor and efficient hardening for histogram of oriented gradients", Proceedings of the IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, Cambridge, UK, 23-25 October 2017, pp. 1–6.

[20] C. Lunardi, F. Previlon, D. Kaeli, P. Rech, "On the Efficacy of ECC and the Benefits of FinFET Transistor Layout for GPU Reliability", IEEE Transactions on Nuclear Science, Vol. 65, No. 8, 2018, pp. 1843–1850.

[21] F. F. dos Santos, L. Draghetti, L. Weigel, L. Carro, P. Navaux, P. Rech, "Evaluation and mitigation of soft-errors in neural network-based object detection in three gpu architectures", Proceedings of the 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops, Denver, CO, USA, 26-29 June 2017, pp. 169–176.

[22] Y. Ibrahim et al., "Soft Error Resilience of Deep Residual Networks for Object Recognition", IEEE Access, Vol. 8, 2020, pp. 19490–19503.

[23] A. P. Twinanda, S. Shehata, D. Mutter, J. Marescaux, M. De Mathelin, N. Padoy, "EndoNet: A Deep Architecture for Recognition Tasks on Laparoscopic Videos", IEEE Transactions on Medical Imaging, Vol. 36, No. 1, 2017, pp. 86–97, 2017.

[24] A. Jin et al., "Tool detection and operative skill assessment in surgical videos using region-based convolutional neural networks", Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Lake Tahoe, NV, USA, 12-15 March 2018, pp. 691–699.

[25] M. Grewal, M. M. Srivastava, P. Kumar, S. Varadarajan, "RADnet: Radiologist level accuracy using deep learning for hemorrhage detection in CT scans", Proceedings of the International Symposium on Biomedical Imaging, Washington, DC, USA, 4-7 April 2018, pp. 281–284.

[26] J. A. Dunnmon, D. Yi, C. P. Langlotz, C. Ré, D. L. Rubin, M. P. Lungren, "Assessment of convolutional neural networks for automated classification of chest radiographs", Radiology, Vol. 290, No. 3, 2019, pp. 537–544.

[27] Z. Wang, A. M. Fey, "Deep learning with convolu-

tional neural network for objective skill evaluation in robot-assisted surgery", International Journal of Computer Assisted Radiology and Surgery, Vol. 13, No. 12, 2018, pp. 1959–1970.

[28] J. H. Ko, B. Mudassar, T. Na, S. Mukhopadhyay, "Design of an Energy-Efficient Accelerator for Training of Convolutional Neural Networks using Frequency-Domain Computation", Proceedings of the Design Automation Conference, Austin, TX, USA, 18-22 June 2017.

[29] A. Azizimazreah, Y. Gu, X. Gu, L. Chen, "Tolerating Soft Errors in Deep Learning Accelerators with Reliable On-Chip Memory Designs", Proceedings of the IEEE International Conference on Networking, Architecture and Storage, Chongqing, China, 11-14 October 2018, pp. 1–10.

[30] H. Alemzadeh, J. Raman, N. Leveson, Z. Kalbarczyk, R. K. Iyer, "Adverse events in robotic surgery: A retrospective study of 14 years of fda data", PLoS ONE, Vol. 11, No. 4, 2016, pp. 1–20, 2016.

[31] Nvidia, "750 Ti White Paper", 2014. (Online: http://international. download. nvidia. com/geforce-com/international/pdfs/GeForce-GTX-750-Ti-Whitepaper. Pdf).

[32] R. C. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies", IEEE Transactions on Device and materials reliability, Vol. 5, No. 3, 2005, pp. 305–316.