UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

RUBENS LUIZ RECH JUNIOR

# Reliability of Google's Tensor Processing Units for Embedded Applications

Work presented in partial fulfillment
of the requirements for the degree of
Bachelor in Computer Engineering

Advisor: Prof. Dr. Paolo Rech

Porto Alegre
May 2022

*“I must be willing to give up what I am*
*in order to become what I will be.”*
— Albert Einstein

## ACKNOWLEDGEMENTS

I could not start expressing my gratitude for anyone other than my Professor Paolo Rech. He has been, by far, the best professor I have ever had who, of course, was fundamental for the accomplishment of this work. Although I did not have the opportunity to have college classes with him, he has been my scientific initiation advisor for over three years already! He not only taught me a lot technically and academically, but he always pushed me further, motivated me to learn more and go beyond what I thought was my limit. I am immensely grateful for everything, and I am profundly proud of being his student. Besides that, although separated by an ocean, the weekly meetings and calls during these incredible three years have made us close and I am glad to call him a very important friend of mine.

To the most important people in my life, my mom and dad, there are not enough words to show how grateful I am for all the motivation and support they have given me throughout my life.For all the happy moments together, the access to countless opportunities that changed my life, and unconditional love, I will be forever grateful to them. They always made me believe in myself, taught me the most important values in life and, for all that, I became who I am today. And I am immeasurably proud of who I am and the fathers I have!

My sincere thanks to my girlfriend Letícia, who was very important for this work, not only because she helped me to review it with very wise opinions, but she showed me the most beautiful feelings and, with that, she genuinely inpired me. She has taught me a lot about life, especially making me happier than ever.

I would like to express my deep gratitude to the most important friends of mine, Filipe, Guilherme, Rafael and Thomaz (alphabetically ordered!). We have been through the craziest and most amazing experiences together, as well as the funniest moments. They inspire me to find balance in me and encourage me to seek and be the best version of myself. I thank Guilherme, in particular, who supported me and was a fundamental person especially during my first year at the university.

Last but not least, I would like to acknowledge the precious effort of the ChipIR and ILL teams. Thanks to them, either remotely or in person, we were able to carry out the radiation experiments that are the foundation of this work.

## AGRADECIMENTOS

Eu não poderia começar a expressar minha gratidão a ninguém que náo o meu professor Paolo Rech. Ele tem sido, de longe, o melhor professor que já tive e, claro, foi fundamental para a realização deste trabalho. Apesar de não ter tido a oportunidade de ser aluno dele em disciplinas na faculdade, ele tem sido meu orientador de iniciação científica há mais de três anos! Ele não só tem me ensinado muito tecnicamente e academicamente, mas sempre me motivou a aprender mais e ir além do que eu achava ser o meu limite. Sou imensamente grato por tudo, e tenho muito orgulho de ser seu aluno. Além disso, embora separados por um oceano, as reuniões e ligações semanais durante esses três incríveis anos nos aproximaram e tenho o prazer de chamá-lo de um amigo muito importante.

Às pessoas mais importantes da minha vida, minha mãe e meu pai, não há palavras suficientes para demonstrar o quanto sou grato por toda a motivação e apoio que me deram ao longo da minha vida. Por todos os momentos felizes juntos, o acesso a inúmeras oportunidades que mudaram minha vida e o amor incondicional, serei eternamente grato a eles. Sempre me fizeram acreditar em mim mesmo, me ensinaram os valores mais importantes da vida e, por tudo isso, me tornei quem eu sou hoje. E estou imensamente orgulhoso de quem sou e dos pais que tenho!

Meus sinceros agradecimentos à minha namorada Letícia, que foi muito importante para este trabalho, não só porque me ajudou a revisá-lo com opiniões muito sábias, mas me mostrou os mais lindos sentimentos e, com isso, me inspirou genuinamente. Ela tem me ensinado muito sobre a vida, especialmente me tornando mais feliz do que nunca.

Gostaria de expressar a minha profunda gratidão aos meus mais importantes amigos, Filipe, Guilherme, Rafael e Thomaz (por ordem alfabética!). Passamos juntos pelas experiências mais loucas e incríveis, assim como pelos momentos mais engraçados. Eles me inspiram a encontrar equilíbrio em mim e me incentivam a buscar e ser a minha melhor versão. Agradeço ao Guilherme, em especial, que me apoiou e foi uma pessoa fundamental principalmente durante meu primeiro ano na universidade.

Por último, mas não menos importante, gostaria de reconhecer o precioso esforço das equipes do ChipIR e do ILL. Graças a eles, remotamente ou pessoalmente, pudemos realizar os experimentos de radiação que fundamentam esse trabalho.

## ABSTRACT

Convolutional Neural Networks (CNNs) have become the most used and efficient way to identify and classify objects in a scene. CNNs are today fundamental not only for autonomous vehicles, but also for Internet of Things (IoT) and smart cities or smart homes. Vendors are developing low-power, extremely efficient, and low-cost dedicated accelerators to allow the execution of the computational-demanding CNNs even in applications with strict power and cost budgets.

In this work we investigate the reliability of Google's Coral Tensor Processing Units (TPUs) to both high-energy atmospheric neutrons (at ChipIR) and thermal neutrons from a pulsed source (at EMMA) and from a reactor (at TENIS). We report data obtained with an overall fluence of $3.41 \times 10^{12} n/cm^2$ for atmospheric neutrons (equivalent to more than 30 million years of natural irradiation) and of $7.55 \times 10^{12} n/cm^2$ for thermal neutrons.

We evaluate the behavior of TPUs executing elementary operations with increasing input sizes (standard convolutions or depthwise convolutions) as well as eight CNNs configurations. Regarding the CNNs, we consider four well-known and widely-used network architectures (SSD MobileNet v2, SSD MobileDet, Inception v4 and ResNet-50) trained with popular datasets, such as COCO and ILSVRC2012. Through retraining, we also assess the impact of transfer learning and a reduced number of object classes to be detected/classified on the CNN prediction robustness.

We found that, despite the high error rate, most neutrons-induced errors only slightly modify the convolution output and do not change the CNNs detection or classification. By reporting details about the error model we provide valuable information on how to design the CNNs to avoid neutron-induced events to lead to miss detections or classifications.



**Keywords:** Artificial Intelligence. Convolutional Neural Networks. Machine Learning. Embedded Applications. Tensor Processing Units. Radiation Experiment. Reliability.

## RESUMO

Redes neurais convolucionais (CNNs) têm se tornado a maneira mais utilizada e eficiente de identificar e classificar objetos em uma cena. Hoje, as CNNs são fundamentais não apenas para os veículos autônomos, mas também para aplicações relacionada a Internet of Things (IoT), casas e cidades inteligentes. Fabricantes estão desenvolvendo acelaradores dedicados extremamente eficientes, de baixa potência e baixo custo para permitir a execução de CNNs de alta demanda computacional mesmo em aplicações com rigorosos orçamentos de energia e custos.

Neste trabalho, investigamos a confiabilidade da Google Coral Tensor Processing Units (TPUs) a nêutrons atmosféricos de alta energia (no ChipIR) e nêutrons térmicos gerados por uma fonte pulsada (no EMMA) e por um reator (no TENIS). Reportamos dados obtidos com um fluência média de $3.41 \times 10^{12}\ n/cm^2$ para nêutrons atmosféricos (equivalente a mais de 30 milhões de anos de irradiação natural), e de $7.55 \times 10^{12}\ n/cm^2$ para nêutrons térmicos. Avaliamos o comportamento das TPUs executando operações elementares (convolução *standard* e convolução *depthwise*) com tamanhos de entrada crescentes, bem como oito configurações de CNNs. Com relação às CNNs, consideramos quatro arquiteturas de redes conhecidas e amplamente utilizadas (SSD MobileNet v2, SSD MobileDet, Inception v4 e ResNet-50) treinadas com *datasets* populares, como COCO e ILSVRC2012. Por meio do retreinamento, também analisamos o impacto da técnica de *transfer learning* e de um número reduzido de classes de objetos a serem detectadas/classificadas na robustez da predição da CNN.

Descobrimos que, apesar da alta taxa de erros, a maioria dos erros induzidos por nêutrons modifica apenas ligeiramente a saída da convolução e não altera o resultado da classificação/detecção. Ao reportar detalhes a respeito do modelo de erros, fornecemos informações valiosas sobre como projetar CNNs de maneira a evitar que eventos induzidos por nêutrons levem a erros de classificação/detecção.



**Palavras-chave:** Inteligência Artificial, Redes Neurais Convolucionais, Aprendizado de Máquina, Aplicações Embarcadas, Unidades de Processamento Tensoras, Experimento de Radiação, Confiabilidade.

## LIST OF ABBREVIATIONS AND ACRONYMS

| | |
|---|---|
| AI | Artificial Intelligence |
| CNN | Convolutional Neural Network |
| CPU | Central Processing Unit |
| DepthConv | Depthwise Convolution |
| DNN | Deep Neural Network |
| DUE | Detected Unrecoverable Error |
| ECC | Error Correction Code |
| EMMA | Equipment Materials and Mechanics Analyzer |
| FIT | Failure In Time |
| FP16 | 16-bit Floating Point |
| FP32 | 32-bit Floating Point |
| FPGA | Field Programmable Gate Arrays |
| GPU | Graphic Processing Unit |
| ILL | Institute Laue-Langevin |
| IoT | Internet of Things |
| ISA | Instruction Set Architecture |
| ML | Machine Learning |
| NN | Neural Network |
| PAC-G | Platform for Advanced Characterisation of Grenoble |
| PCIe | Peripheral Component Interconnect Express |
| RAL | Rutherford Appleton Laboratory |
| RGB | Red-Green-Blue |
| TENIS | Thermal and Epi-thermal Neutron Irradiation Station |
| TL | Transfer Learning |

| | |
|---|---|
| TOPS | Tera Operations Per Second |
| TPU | Tensor Processing Unit |
| SDC | Silent Data Corruption |
| SSD | Single-Shot multibox Detection |
| StdConv | Standard Convolution |
| UINT8 | 8-bit Unsigned Integer |
| UK | United Kingdom |
| USB | Universal Serial Bus |

## LIST OF FIGURES

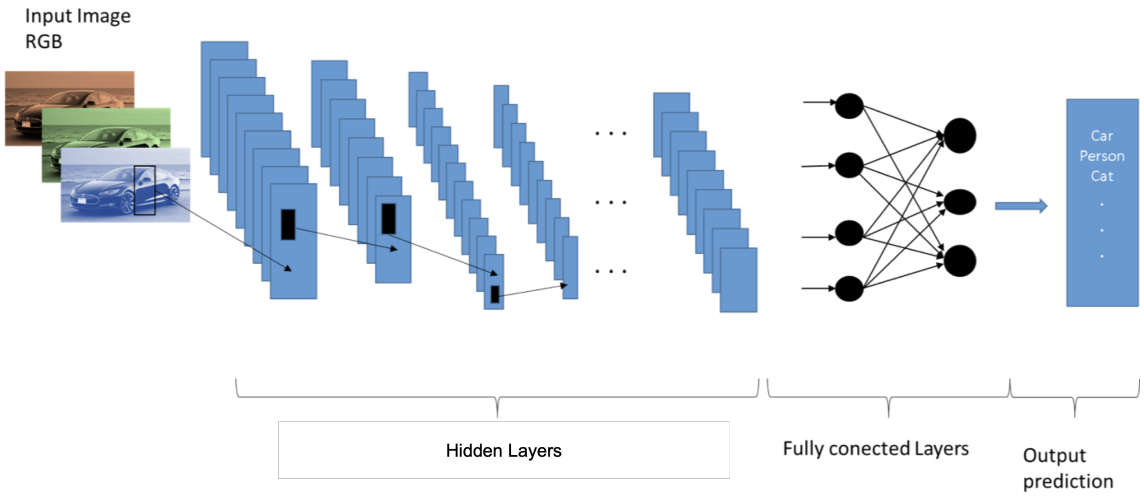
## LIST OF TABLES

**CONTENTS**

# 1 INTRODUCTION

Convolutional Neural Networks (CNNs) are today the most effective (and efficient) way to detect an object in a scene. By applying various filters to the input image, convolutional layers extract information into feature maps that are then passed to the downstream layers to detect and/or classify objects. The number of layers, the kind of filter applied, and several other hyper parameters that define the structure of the CNN are engineered to achieve the desired efficiency and accuracy. Figure 1.1 shows a simplified scheme of the structure of a CNN.

Figure 1.1: Structure of a CNN for object detection




The prediction process is highly computational demanding, as it is necessary to apply several filters to each feature map. When it comes to hardware and software implementation, the filtering process is mapped into a matrix multiplication operation, which can be efficiently executed in parallel accelerators, such as Graphics Processing Units (GPUs) or Field Programmable Gate Arrays (FPGAs).

To ensure very high accuracy along with real-time detection (at least 40 frames per seconds must be processed), both being fundamental for applications like autonomous vehicles, it is necessary to execute CNNs on highly performant, costly, and power-hungry devices, such as the latest GPUs or very big FPGAs. Nevertheless, the field of adoption of CNNs is not limited to self-driving cars. Many other applications, with less strict accuracy and timing constraints, can benefit from CNNs execution. This is the case of Internet of Things (IoT), smart homes and smart cities, in which detecting or identifying a relatively low number of objects can significantly improve the overall system features and, ultimately, enhance the user experience. In these applications, cost and power consumption

must be minimized, while still guaranteeing sufficient prediction accuracy.

In order to meet the requirements of such applications, lately, vendors have developed low-cost accelerators for CNNs execution, named *EdgeAI* devices, such as NeuroShield or Google Coral Tensor Processing Units (TPU). These EdgeAI devices are designed to execute elementary operations (i.e., convolutions and some other matrices operations) in low data precision, i.e., 16-bit floating point or even 8-bit integer. Coupled with a good software framework (e.g., Tensor Flow) that runs on a host device, EdgeAI devices significantly reduce the time and power consumption of the convolution, which is the most computational demanding operation of CNNs.

As EdgeAI devices are likely to be used at scale and in distributed systems, it is fundamental to investigate their reliability, in particular their neutron-induced error rate. Preliminary studies showed that, despite being small, EdgeAI devices have a non-negligible neutrons- or protons-induced error rate (BLOWER et al., 2021; BREWER et al., 2020).

In this work, we investigate the neutrons reliability of the Google Coral TPU by irradiating the device with such particles during radiation beam experiments. Unlike previous works on EdgeAI reliability, we deeply investigate the device fault model on the main elementary operations (standard and depthwise convolutions). Moreover, we compare the error rate and the prediction failures of eight CNNs configurations that are widely used in embedded applications: Single-Shot multibox Detection (SSD) MobileNet v2 and SSD MobileDet, trained with COCO dataset, as well as Inception v4 and ResNet-50, trained with ILSVRC2012 dataset.

To have a broad evaluation, we test the Coral TPU with both high-energy neutrons, at the ChipIR facility, and with thermal neutrons, at the EMMA facility in UK and at TENIS facility at Institut Laue-Langevin (ILL) in Grenoble, France. In order to be able to compare the data obtained with different types of neutrons, we report experimental results using a metric called *cross section* which, ultimately, represents the probability of an energetic particle to induce an error in the program execution (more details are provided in Section 2.3). We observe that, while the high-energy neutrons cross section of the Coral TPU is much higher than the thermal neutrons cross section, the results are consistent in the sense that depthwise convolutions are shown to have higher error rate than standard convolutions and SSD MobileDet is less reliable than SSD MobileNet V2.

The rest of the text is organized as follows. In Chapter 2, we provide a solid background on CNNs and their reliability to transient faults in other architectures and

devices, as well as the hardware and software architecture of Coral TPU, useful for understanding experimentally observed behaviors. In Chapter 3, we describe the thermal and high-energy neutron setups we developed and the software (convolutions and CNNs) we tested. Experimental results are presented and discussed in Chapter 4, highlighting the implications for future hardening solutions for Coral TPU, while Chapter 5 concludes the document.

# 2 BACKGROUND

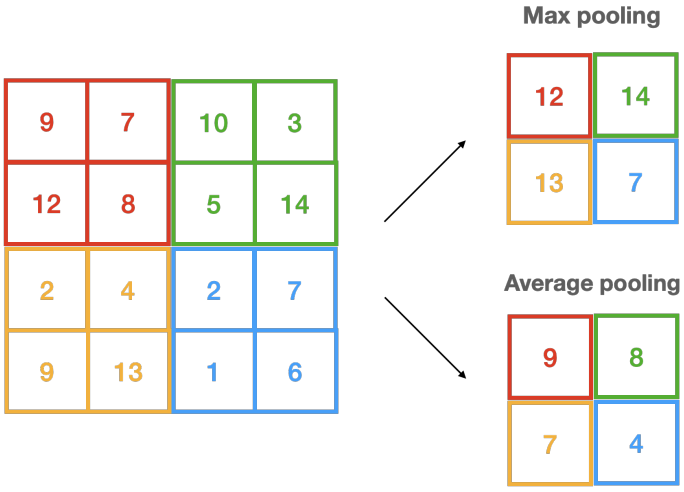
In this Chapter, we review the main characteristics of CNNs, the architecture of EdgeAI devices (focusing on the Coral TPU), the software framework used to train and execute CNNs on EdgeAI accelerators and we introduce some metrics for measuring the device reliability.

## 2.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are today widely adopted to perform object detection (REDMON et al., 2015). A CNN is a sequence of layers of different kind, each applying a specific function to the input frame or input feature map. Among several types of layers, the most common and fundamental ones in modern Deep Neural Networks (DNNs) are: convolutional layers, pooling layers and fully connected layers.

Pooling layers are used to down sample the dimensions of feature maps by applying functions that summarize the features in each block of the feature map that was extracted by convolutional layers. Average pooling and max pooling are the most common functions and they respectively represent the average and the most activated features from the given input feature map into its output.

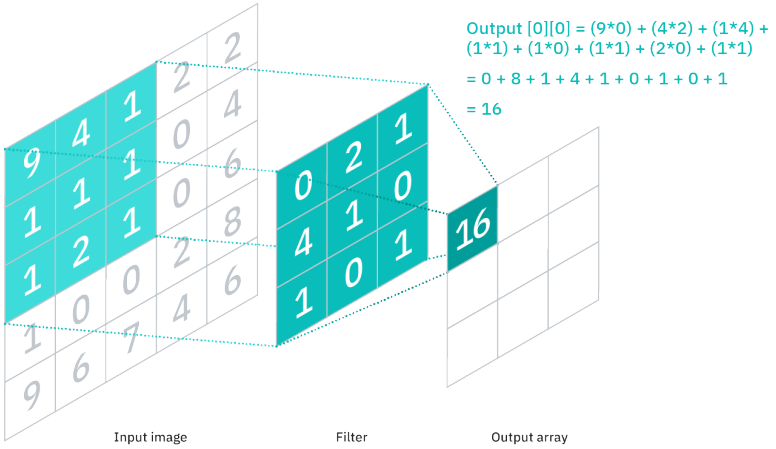
Figure 2.1: Pooling layer of size 2x2 and stride 2




Fully connected layers are conventional neural networks that are in charge of performing the object predition task. They provide the probabilities, classes and positions of each possibly detected object based on the features that were extracted by the upstream layers in the chain (essentially by convolutional layers).

Given this, it is clear that one of the main steps when using CNNs for object detection is *convolution*. Convolutional layers are the computational core of CNNs. By applying filters, they are responsible for extracting information, from the input frame, which is then processed to identify objects. To extract specific characteristics of the image, a kernel filter is convolved with a matrix, i.e., the kernel slides over the input matrix, multiplying and accumulating products at every position of the input with every position of the kernel. More than 80% of the computation in a CNN is dedicated to convolution, which is why most device architects are focusing on making convolution more and more efficient, producing novel devices such as the Coral TPU.

Figure 2.2: Convolutional layer example. Source: (IBM Cloud Education, 2020)




Lately, it has been shown that the efficiency of CNNs execution can be significantly improved by approximating operations (Hanif et al., 2018) or hardware components (Sarwar et al., 2018; MRAZEK et al., 2016), and it has been also demonstrated that the same object detection accuracy can be achieved, through re-training, representing data in 16-bit floating-point (GUPTA et al., 2015), 8-bit integer, or even in binary values (GAMBARDELLA et al., 2019). Therefore, most low-power accelerators take advantage of reduced-precision operations to reduce the computing power required to run CNNs. The Coral TPU we used in this study, for instance, executes operations in 8-bit integer.

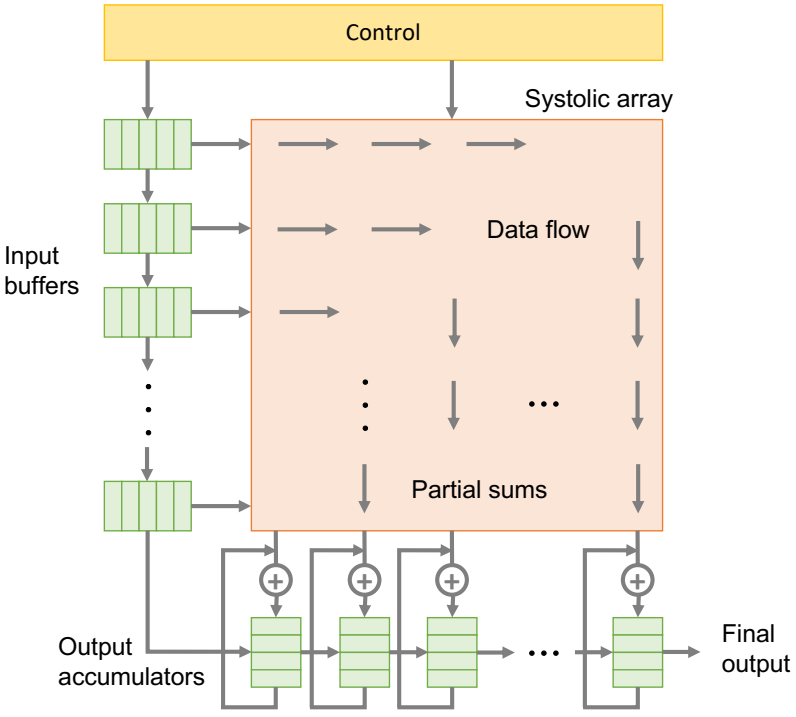
## 2.2 EdgeAI accelerators

EdgeAI accelerators, like NeuroShield and Google Coral TPU, are low-power and low-cost devices designed to perform heavy machine learning computations in the context

of embedded applications. As it is the main target of this study, we focus our analysis on the Google TPU.

Figure 2.3 shows the high level schematic of the Coral architecture which is mainly composed by a systolic array fed by a large set of input buffers. Because it is a low-cost device, these buffers are not protected by error-correction code (ECC) technology, which would make the hardware more resilient to transient faults, however would increase the manufacturing cost. The systolic array outputs the product of the model weights and each layer's input into the activation unit, where the partial sums are accumulated and the activation function is applied. Therefore, this device can perform a set of operations, mainly convolutions, which are a fundamental block for machine learning applications, in an extremely power- and performance-efficient manner. As a metric that indicates the power efficiency, the TPU delivers 2 TOPS (tera operations per second) per watt.

Figure 2.3: High level schematic of the Coral Edge TPU architecture. Adapted from (Q-ENGINEERING, 2019).




Since Coral TPU is simply an accelerator, it must be connected to a host device. Google provides two versions of the accelerator: one that interfaces with the host via PCIe and the other uses USB 3.0. On our setup, we have a Raspberry Pi 4 as host, connected to the Coral USB accelerator. For minimizing data transfers and storage and speed up calculations, all data that is computed and stored within the TPU is represented as 8-bit unsigned integers (UINT8). The device is capable of performing the quantization and

de-quantization steps for interfacing with the host floating-point representations.

The software layer of the Coral TPU relies on TensorFlow Lite which is a light and optimized-for-embedded-devices version of the TensorFlow framework that was developed by Google for machine learning (ABADI et al., 2015). Most of the development effort is very similar as if the machine learning (ML) model would run on a normal central processing unit (CPU), however there is an EdgeTPU compiler that is responsible for deploying the TensorFlow Lite model targeting the Coral Edge TPU architecture.

## 2.3 Reliability Indicators and Metrics

One of the most common metrics to indicate the reliability of a device is called *Failure in Time* (FIT). It is an industry standard value that represents the number of failures/errors per billion hours of operation.

Another frequently used metric for this purpose is the *cross section*, measured as area [$cm^2$], which stands for the area of the device that, when hit by an energetic particle, will generate a fault. Thus, the cross section represents the probability of radiation-induced errors to occur.

In this work, however, we use the cross section as the standard way to report reliability of the Coral TPU of each software workload we test. The main reason is that the sensitivity to thermal neutrons is strictly related to the amount of Boron-10 used in the device production, which is normally a business sensitive information not available to the public, and, as observed in (OLIVEIRA et al., 2021), the exact flux of thermal neutrons that hits the device board (needed to calculate the FIT rate) depends on various factors, such as the humidity of the air and the interaction of the hardware chip with the surrounding materials. Therefore, it is not possible to provide accurate FIT rates for thermal neutrons and we use the cross section to be able tocompare results between experiments with different types of particles.

FIT rates for experiments performed with atmospheric neutrons (at ChipIR) are reported to give an idea of the average number of errors that would occur under natural conditions with TPU Coral exposed to natural terrestrial irradiation.

## 2.4 CNNs Reliability

CNNs have already been shown to be particularly susceptible to transient faults in many studies (Santos et al., 2019; BOSIO et al., 2019). Through radiation beam experiments and fault-injection, it has been demonstrated that the corruption of each layer has a different probability of affecting the CNN output, with the convolutional layers being responsible for most observed errors (Santos et al., 2019).

The corruption of a layer or an operation inside a layer can be:

- *masked* without affecting the output;
- reach the output but keep the classification/detection unaltered, characterizing a *Silent Data Corruption (SDC)*;
- spread and modify the output in ways that impacts the functionality and outcome of the CNN, leading to a *critical SDC*;
- affect the software control logic and result in application crash, which is called *Detected Unrecoverable Error (DUE)*.

Thanks to the intrinsic approximate nature of CNN computation, most of the errors do not turn into system failures, i.e., they do not affect the CNN accuracy. This has been proven for GPUs (Santos et al., 2019), FPGAs (LIBANO et al., 2018), and NeuroShield devices (BLOWER et al., 2021; BREWER et al., 2020). Unfortunately, despite the approximate inherent nature, the misdetections and misclassifications rates in CNN executed in modern computing devices are still too high to be employed in safety-critical applications (Santos et al., 2019; BOSIO et al., 2019). As discussed in Section 4.2.2, we distinguish between critical and tolerable errors in CNN execution on the Coral TPU. Additionally, we investigate the distribution of corrupted element at the output of convolutions (results in Section 4.1.2).

As already mentioned, Coral TPU executes operations in 8-bit unsigned integers to improve performance. It has been shown that reducing operation data-precision, while bringing unquestionable benefits to efficiency, has the drawback of increasing the (negative) impact of a fault on the operation output (Fernandes dos Santos et al., 2019). For CNNs, precision reduction turns into a higher probability of a fault to modify the detection. It has been demonstrated that a fault in a FP16 CNN has $\sim$2x the probability of causing misdetection than a fault in a FP32 CNN (LIBANO et al., 2020). For that reason, part of our contribution is to evaluate whether the execution of CNNs using 8-bit integer

is harmful for the system reliability.

Recently, some works have discussed the reliability of EdgeAI devices to neutrons and protons, focusing specifically on the Arduino NeuroShield (BLOWER et al., 2021; BREWER et al., 2020). To the best of our knowledge, this is the first work presenting experimental data on Coral TPU devices error rate. Previous studies have shown that the error rate of the small EdgeAI accelerators is far from being negligible (higher than $10^2$ Failure In Time - FIT rates).

Unlike previous publications, we engineered an experiment setup to test not only neural networks but also atomic operations performed by the accelerator (convolutions) with different sizes and depths (2D and 3D). This information is useful to deeply investigate the neutron-induced fault model of the TPU.

# 3 METHODOLOGY

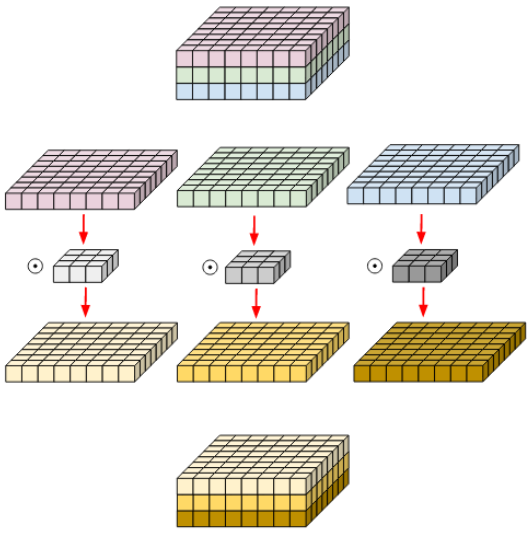
In this Chapter, we detail the software we run, the hardware setup we prepared and the (high-energy and thermal) neutron beam experiments we performed.

## 3.1 Software layer

### 3.1.1 Convolutions

Coral Edge TPU is designed to accelerate machine learning algorithms, especially neural networks. Considering that most of the computating effort of deep neural networks is fundamentally represented by convolution operations, we can say the basic operation of a Coral TPU is indeed convolution. Besides that, from the instruction set architecture (ISA) perspective, convolutions are atomic operations.

Figure 3.1: Visual representation of the depthwise convolution algorithm. Source (PANDEY, 2018)



Hence, as a first experiment, we want to evaluate the reliability of the two types of convolution that are supported by Coral: standard and depthwise. *Standard convolutions* are normal 2D convolutions while *depthwise convolutions* have an input composed by multiple channels and each one is convolved with its respective kernel separately, as shown in Figure 3.1. Since CNNs usually perform image prediction, in our experiments, the inputs of depthwise convolutions are always composed of three channels, as for the RGB colors, and this type is referred as 3D convolutions. We run tests with squared ma-

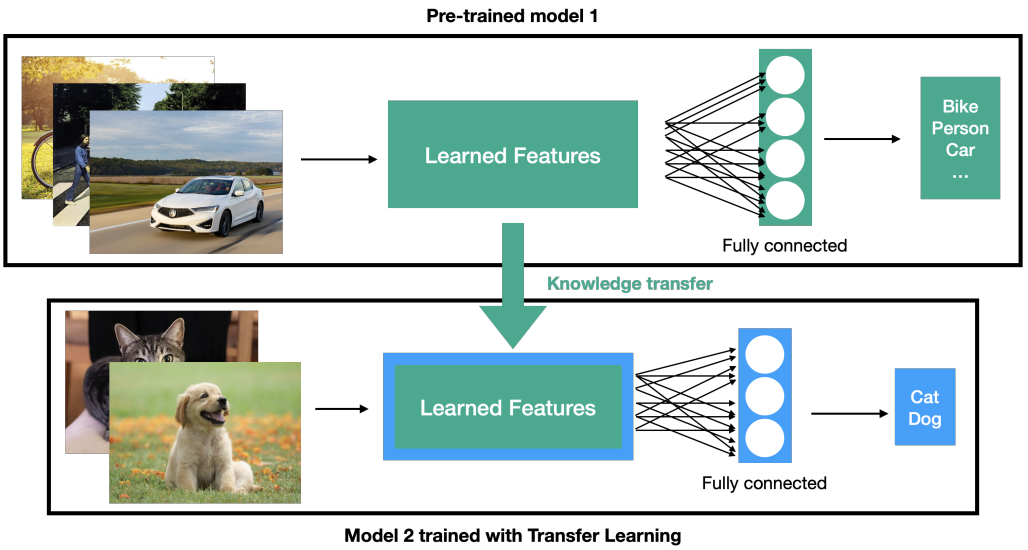
trixes of sizes ranging from 256 to 1,250 (INT8) as inputs and squared kernels of fixed sizes: 40 for standard convolutions and 20 for the depthwise ones.

### 3.1.2 Convolutional Neural Networks

Besides convolutions, we evaluate the reliability of eight convolutional neural network configurations in which we vary the network architecture, the dataset, and the training methodology. We consider CNNs that perform the two main machine learning tasks supported by Coral: image classification and object detection. In image classification, the goal is to classify a single object in the image, e.g., a dog, a car or a tree, without indicating the position. On the other hand, object detection is a more complex task, as multiple objects in the image need to be located and then classified.

For a broad assessment, we consider four different network architectures. Two of them, Inception V4 (SZEGEDY et al., 2017) and ResNet-50 (HE et al., 2016), target image classification. Both are trained with ILSVRC (RUSSAKOVSKY et al., 2015) dataset and support a wide range of 1,000 different object classes. The other two, SSDLite MobileDet (XIONG et al., 2020) and SSD MobileNet V2 (SANDLER et al., 2018), perform object detection and are trained with COCO (LIN et al., 2014) dataset which embraces 90 classes. The models for these NNs are based on TensorFlow Lite – the machine learning framework developed by Google for embbeded applications and especially optimized for Coral Edge TPU.

Figure 3.2: Transfer learning reuses knowledge from pre-trained model during training of a new model



In addition to these four models/configurations, we also retrain SSD MobileNet V2 with two other datasets: a subset of the COCO dataset, containing 14 object classes, and a subset of the Oxford-IIIT Pet (PARKHI et al., 2012) dataset with only 2 classes. Our goal is to evaluate whether and how a reduced number of objects to be detected impacts the device error rate.

The retraining process is done with and without the application of *transfer learning* technique. When applying transfer learning, the knowledge from a pre-trained machine learning model is reused during the training of the new model in order to speed up the learning process and, in most cases, this even improves classification/detection performance. As represented in Figure 3.2, most of the feature extractor is reused from the pre-trained model and the training of the new model is basically reduced to adjusting weights of the neurons in the fully connected layers.

Considering both types and multiple sizes of convolutions, as well as the different CNN configurations, we provide experimental data obtained on 16 benchmarks.

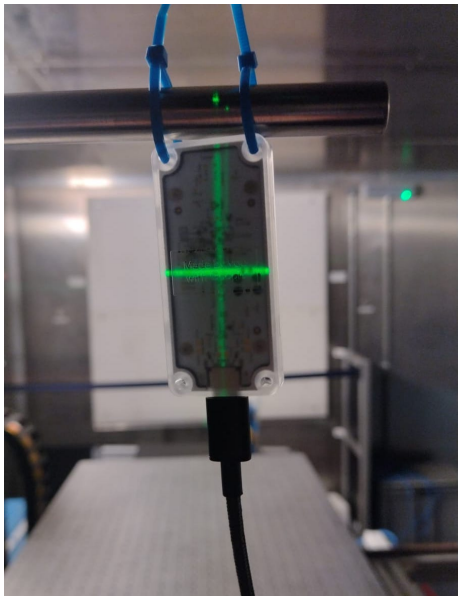
## 3.2 Neutron Experiment Setups

### 3.2.1 High-energy Neutrons at ChipIR

The atmospheric neutron experiments were carried out at the ChipIR facility at the ISIS spallation neutron source of the Rutherford Appleton Laboratory (RAL), UK. ChipIR (CAZZANIGA; FROST, 2018) is the reference beamline dedicated to the irradiation of microelectronics and it features a high-energy neutron spectrum, as similar as possible to the atmospheric one. The flux with neutron energy above 10 MeV is $5.4 \times 10^6 n/cm^2/s$, while the thermal component ($E < 0.5eV$) is $4 \times 10^5 n/cm^2/s$ (CHIESA et al., 2018).

The Coral TPU was positioned 0.8 meters away from the ChipIR beam-stop with a collimated beam size of $70 \times 70$ mm. A picture of the Coral TPU at ChipIR is shown in Figure 3.3. At the Coral position, the average flux was about $3.9 \times 10^6 n/cm^2/s$. The host device, a Raspberry Pi 4, is connected with a 2-meters-long USB cable and placed well outside the beam. We test the device for more than 241 effective hours, already excluding the time spent loading the input images to the memory, downloading the output matrices to the host device, and rebooting the devices when necessary (some Detected Unrecoverable Errors may lead to system reboot). The resulting neutrons fluence was then greater than

Figure 3.3: The Coral TPU aligned with the high-energy neutron beam at ChipIR



$3.41 \times 10^{12} n/cm^2$. and, when scaled to the terrestrial flux ($13 n/cm^2/h$ (SLAYMAN, 2010)), this corresponds to more than 30 million years of natural irradiation, in case of a device on the terrestrial surface in sea level.

### 3.2.2 Thermal Neutrons at EMMA

The ISIS neutron source also features various thermal neutrons facilities, such as the Equipment Materials and Mechanics Analyzer (EMMA) (CAZZANIGA et al., 2021) that has a line of sight on the water moderator of the main neutron source. The thermal neutron beam is achieved from the pulsed neutrons source thanks to a *chopper*, i.e., a rotating device used to block a portion of the neutron beam in time, that is synchronous with the proton pulse, thus cutting the fast neutron portion of the spectrum, letting through only the thermal component. The thermal neutron flux delivered at EMMA is of about $2.32 \times 10^6 n/cm^2/s$. More details about the neutrons spectrum and the flux measurements at EMMA can be found in (CAZZANIGA et al., 2021).

The availability of both high-energy (ChipIR) and thermal (EMMA) neutrons facilities at ISIS is very convenient, as the same setup and the same devices can be tested back-to-back in both beam lines, allowing a direct comparisons of the sensitivity of the same device to two different neutrons spectra. Nevertheless, considering that cross sections obtained with thermal neutrons are normally significantly lower than the one obtained with high-energy neutrons, EMMA flux might be too low to test small configura-

tions (applications that do not demand much performance of the device due to reduced amount of data to be processed). With that in mind, we used EMMA to characterize the TPU configurations with the highest error rates (MobDet and MobNet CNNs). After more than 25h of test at EMMA the 1,024 convolutions provided only 10s of SDCs (Silent Data Corruptions), making the characterization impractical in this facility.

### 3.2.3 Thermal Neutrons at TENIS

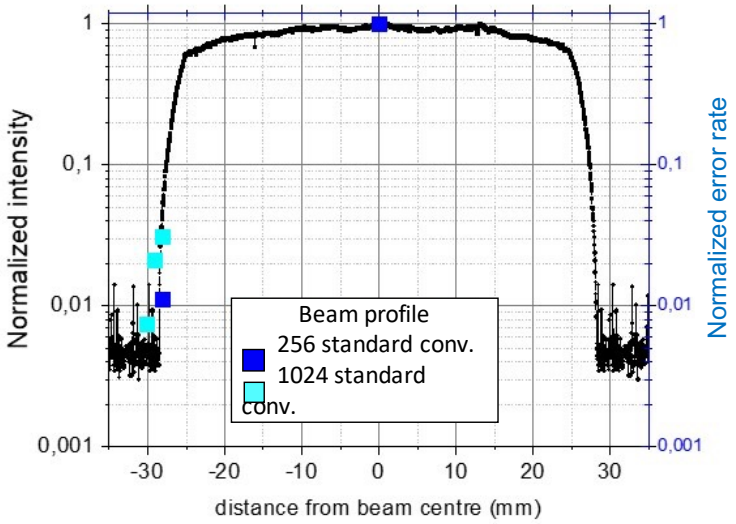
To measure the thermal-neutrons cross section of the TPU executing convolutions (that would not be possible at EMMA due to the very low error rate), we also perform experiments at the new Thermal and Epithermal Neutron Irradiation Station (TENIS) hosted by the Institut Laue-Langevin (ILL). This new facility aims to replace D50 as a facility where thermal neutron experiments were conducted at the Platform for Advanced Characterisation of Grenoble (PAC-G) (BEAUCOUR et al., 2015; WEULERSSE et al., 2018).

A captured flux of $1.92 \times 10^9 n/cm^2/s$ has been measured by Au foil activation. TENIS beam is a 5x5 $cm^2$ square. As shown in Figure 3.4, the flux is very stable in a 2x2 $cm^2$ centered square, and from there it decreases rapidly. The sample was tested initially in the middle of the beam spot where the flux is well characterized. In that position the error rate is so high that in a few hours we observed more than 100 SDCs for the smallest convolution configuration, which has an input matrices of 256x256. So,because of the high flow in the center, it is not possible to test larger configurations. The high flux from the central position was also problematic as after a few hours the devices died, probably due to the gamma rays induced Total Ionizing Dose, and we could no longer get it to work.

Considering the flux at the center is too high, we have then shifted the device to the edge of the beam, moving it from 2.7cm to 3.1cm from the center, with steps of 1mm, in order to reduce the error rate. According to the horizontal beam profile shown in Figure 3.4, the flux significantly drops starting at 2cm from the center, being approximatively $1.4 \times 10^7 n/cm^2/s$ at 3cm from the beam center.

As shown in the Figure 3.4, the error rates obtained for the 256 and 1,024 convolutions with the TPU shifted away from the beam central position, and normalized to the error rate observed at the beam center, follow very well the beam profile measurement. The expected dose rate in SIlicon at TENIS is of about $1,000 Gv/h$ from neutron interac-

Figure 3.4: Cross section of the beam profile at TENIS and the error rate for 256 and 1024 standard convolutions normalized to the error rate measured at the center of the beam



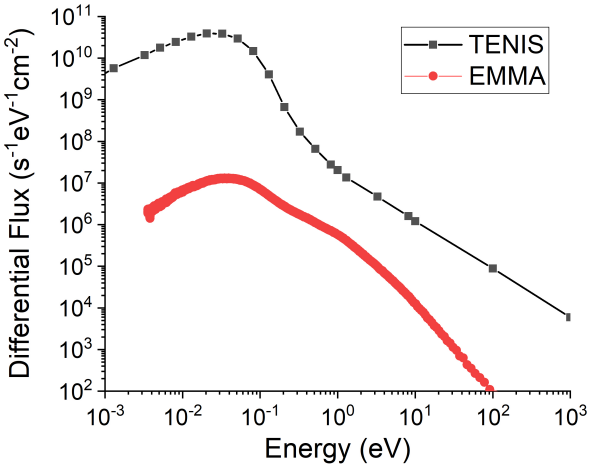

tions and $250Gv/h$ due to gammas coming directly from the reactor. We do not observe any Total Ionizing Dose effect during our experiments.

### 3.2.4 Comparison between TENIS and EMMA

To compare EMMA and TENIS results, normally the EMMA flux is normalized with the 25meV equivalent flux (the peak energy at room temperature). The neutron energy spectra of EMMA and TENIS, shown in Figure 3.5, can be described, in first approximation, as a Maxwell-Boltzmann distribution with a broad peak for thermal neutrons. TENIS, as shown in the plot, has a different (and much higher) spectral contribution of epithermal neutrons than EMMA.

To compare EMMA and TENIS results, we convert the “thermal flux” to “25meV-equivalent flux” (25 meV being the peak-energy at room temperature). The “thermal flux”, as defined in the JESD89A standard (SLAYMAN, 2010) and also as common practice in nuclear physics, is the integrated flux < 0.4 $eV/cm^2/s$. The conversion factor between “thermal flux” and “25meV-equivalent flux” is calculated by integrating the differential flux multiplied by the cross section of B-10 and divided by the cross section of B-10 at 25 meV. The result for EMMA is a factor of 0.71.

Figure 3.5: Comparison of the neutron energy spectra of EMMA and TENIS




### 3.2.5 Notes

All experiments described above are performed at room temperature, using the standard power and frequency configuration of the Coral TPU. We have tested a total of 4 TPUs.

As a consequence of the Covid-19 pandemic situation, experiments in the UK were performed remotely, thanks to the tireless and precious help of the ChipIR team in mounting the setup and granting remote access to the researchers in Brazil and Italy. Experiments in Grenoble were performed in person, which gave to the researchers an optimistic feeling for the close future of radiation experiments.

# 4 EXPERIMENTAL RESULTS

In this Chapter we present data from the neutron experiments obtained by irradiating the TPUs with atmospheric (high-energy) and with thermal (low-energy) neutrons. We consider both Silent Data Corruptions (SDCs, i.e., errors on the output) and Detected Unrecoverable Errors (DUEs, i.e., crashes or hangs). We first discuss the reliability of atomic operations – standard (2D) and depthwise (3D) convolutions – and then the reliability of four different neural networks that were trained with multiple datasets for a total of 8 neural networks configurations.

We recall that we use the cross section value to express and compare the reliability of each benchmark that was tested during the experiments in the three facilities (ChipIR, EMMA and TENIS) with different types and fluxes of neutrons. All data is reported with 95% confidence intervals, considering a Poisson distribution.

## 4.1 Atomic Operations

Aiming to analyze how faults affect the execution of the simplest and most lightweighted operations that the TPU can execute, we run two different types of convolutions: standard and depthwise. We recall that standard convolution stands for normal 2D convolutions while the depthwise convolution algorithm separates the three channels (RGB) of the 3D input matrix and convolves each one with its respective kernel as it was a 2d convolution. In our tests, the inputs of depthwise convolutions always have three channels (as for the RGB colors) and this type is referred as 3D convolutions.

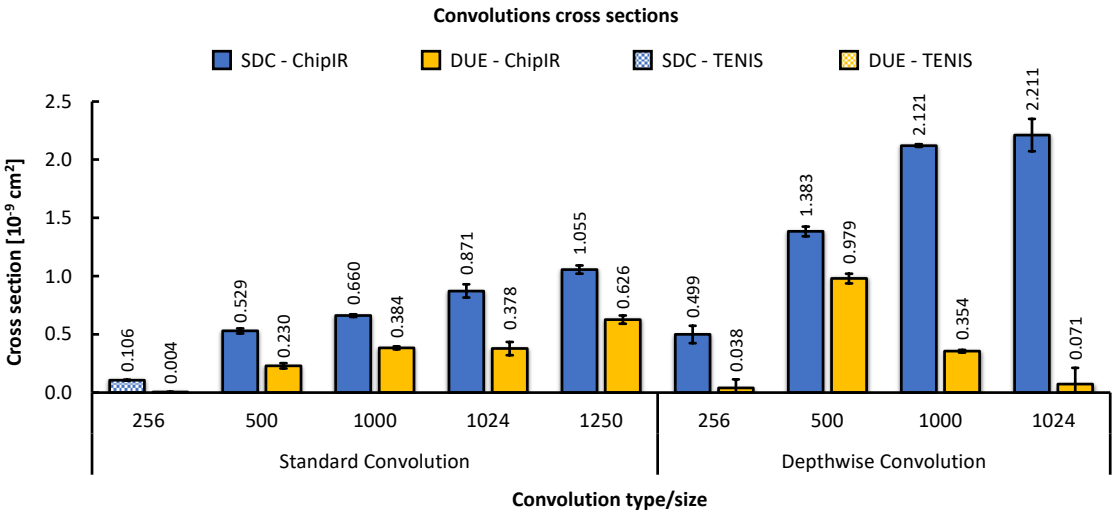
We performed tests with squared matrixes of varying sizes, ranging from 256 to 1,250, as inputs and squared kernels of fixed sizes, 40 for standard convolutions and 20 for the depthwise ones. We choose a kernel size that is both representative (the kernel size is normally much smaller than the feature size) but yet sufficient to saturate the TPU computing capabilities. A kernel of size 40 would exceed the TPU computing capabilities for the 3D convolutions.

### 4.1.1 Error Rates and Cross Sections

Figure 4.1 plots the cross sections (SDCs in blue and DUEs in yellow) for the tested sizes of both convolution types resulting from the high-energy neutron experiments at ChipIR and thermal neutron experiments at TENIS. Due to the low error rate at EMMA (more than 5 hours of experiment was needed to observe *one* error), we decided to test the TPU executing convolutions at TENIS, where the flux is 3 order of magnitude higher, with cold moderation of neutrons.

The results for size 256 of the standard convolution (StdConv) algorithm were obtained at TENIS and are highlighted with a different fill pattern in the left side of the graph. Depthwise convolution (DepthConv) for 1,250 input cannot be executed on the TPU since it exceeds the device computing capabilities.

Figure 4.1: Cross sections for standard and depthwise convolutions, with increasing input sizes, exposed to high-energy neutrons at ChipIR and thermal neutrons at TENIS

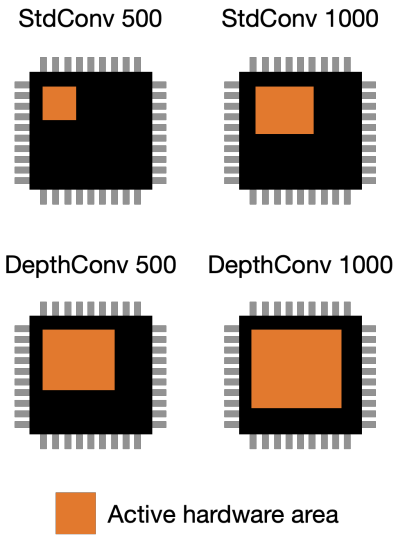



As shown in the Figure above, the SDC cross section increases with the size of the convolution input. Intuitively, this is justified as the systolic array becomes more occupied due to the increasing amount of data that needs to be processed. With more hardware area being actively used, the probability of a neutron hitting the board and causing an error increases.

On the other hand, the DUE cross sections does not follow this trend. This should come as no surprise since, as shown in one of our previous publications (FRATIN et al., 2018), DUEs normally have a component that depend exclusively on the hardware and not the software layer. Thus, DUEs are biased by the sensitivity of hardware resources

and are independent of the executed code (or input size).

Figure 4.2: Illustrative comparison between standard and depthwise convolutions as to hardware resources usage




From Figure 4.1, we also observe that, for a given input size, depthwise convolutions have higher SDC cross section when compared to standard convolutions (on average, 179% higher). In addition, the cross sections of 3D convolutions increase with the input size at a higher rate than those of 2D convolutions. Considering DepthConvs operate on about 3 times more data than StdConvs, this trend is again related to the fact much more area of the TPU device is used when processing depthwise/3D convolutions.

The cross sections for standard convolution of size 256, which were irradiated with thermal neutrons, have the device positioned in the center of the beam at TENIS facility. The flux in this position is too high to test bigger configurations, i.e, convolutions with greater input size. For instance, when compared to the values for StdConv 500 obtained during the experiments with high-energy neutrons at ChipIR facility, the cross section at TENIS is about 5 times smaller. This is in line with previous data on thermal versus high-energy neutrons obtained in various devices (WEULERSSE et al., 2018; OLIVEIRA et al., 2021).

Table 4.1 shows the FIT rates for both types of convolutions when exposed to atmospheric neutrons at ChipIR. We recall that FIT rates can only be calculated for the experiments performed with atmospheric neutrons, therefore, data for the thermal neutron experiment at TENIS cannot be provided.

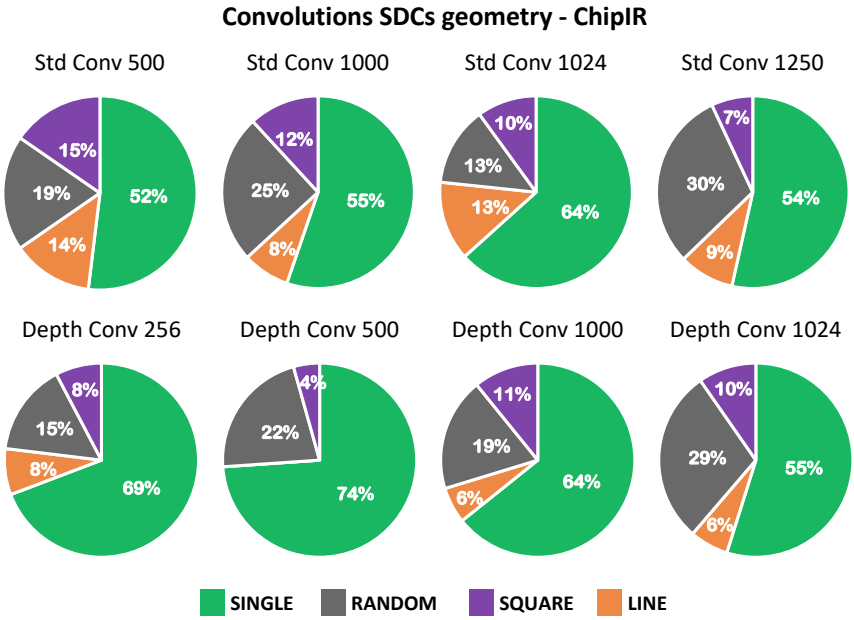
Table 4.1: FIT rates for convolutions exposed to atmospheric high-energy neutrons at ChipIR

| Convolution type | Input size | FIT SDCs | FIT DUEs |
|---|---|---|---|
| Standard Conv | 500 | 6.87 | 2.99 |
| | 1000 | 8.59 | 4.99 |
| | 1024 | 11.33 | 4.91 |
| | 1250 | 13.72 | 8.14 |
| Depthwise Conv | 256 | 6.48 | 0.50 |
| | 500 | 17.98 | 12.73 |
| | 1000 | 27.57 | 4.61 |
| | 1024 | 28.74 | 0.93 |

### 4.1.2 Geometric Distribution of Errors

Figure 4.3 shows the geometric distribution of the output elements that were corrupted during the experiments with convolutions at ChipIR. Data from TENIS is similar and thus not shown.

Figure 4.3: Geometric distribution of the corrupted elements in the output of convolutions




When an SDC is detected, we download the whole output matrix to the Raspberry Pi and identify how many elements in the output are corrupted. When multiple elements are found corrupted, we categorize the corruption based on the spatial distribution of the wrong elements. When more than one element is corrupted and these elements sit in the same row or column, we count a *Line* error. When the corrupted elements are distributed in a square (a whole portion of the output matrix is corrupted), we count a *Square* error.

When we see multiple corrupted elements that are neither on a Line nor on a Square, we count a *Random* error.

It is worth noting that, by tunning the input sizes and flux, we engineered the experiments not to have more than one neutron generating an error in a single execution, since this would be an artifact unlikely for a realistic application. Thus, eventual multiple errors are caused by the spread of the single neutron corruption to multiple operations and not by multiple neutrons corruption.

Regardless of the convolution type, we observe that the distribution is very similar across all sizes. Most of the time, a single element of the output matrix is corrupted. The second most frequent SDC geometry is Square, meaning that the elements corruption occurred within square/rectangular blocks, followed by Random distribution, in which the position of the errors does not match any geometric shape. Finally, element corruptions arranged in a single Line is the least frequent geometric distribution.

The fact that single corrupted elements is the most frequent distribution in the TPU architecture is in contrast with what has been observed for Graphics Processing Units (GPUs) (RECH et al., 2013; Santos et al., 2019; BASSO et al., 2020), for which the majority of the corrupted matrices have multiple corrupted elements. This is due to the different way matrix multiplication is implemented in these architecures. On GPUs, matrix multiplication is executed as a code, with a sequence of instructions, while on the TPU the execution is done as a single instruction in a systolic array. Executing a sequence of instructions, therefore, tend to lead to a higher spread of the error in the output.

As it has been shown that multiple corrupted elements in the output matrix are the main cause for misdetections or misclassifications in convolutional neural networks (Santos et al., 2019), the fact that the TPU is less prone to have multiple output errors than GPUs can be a promising result for its reliability in executing CNNs.

Additionally, we have observed that the magnitude of the errors (i.e., how much the corrupted value is different from the expected one) is, overall, very small. The absolute difference between the expected and the corrupt element value is, in fact, exactly *one* (e.g., the expected value is 80 and the corrupted one is 81 or 79) in 91% of the observed SDCs. Please recall that only INT8 operations can be performed on the TPU.

Also, when the error magnitude is greater than one, the difference with the corrupted and expected value is a power of 2, i.e., a single bit flip usually occurs, and this happens regardless of the convolution type. Again, this is in contrast with data observed for GPUs, for which the magnitude of the error can be significantly higher (orders of

magnitude) (Santos et al., 2019; BASSO et al., 2020) most likely because floating-point representation is used. This is another promising result for the TPU reliability in executing CNNs, as a higher error magnitude can have a greater impact on the output value.

## 4.2 Neural Networks

With regards to neural networks, we report the reliability analysis for eight different configurations by varying the network architecture, dataset and training procedure (with or without transfer learning). We leveraged on four NNs models that were trained and made available by Google (Inception V4, ResNet-50, SSD MobileDet, and SSD MobileNet V2).

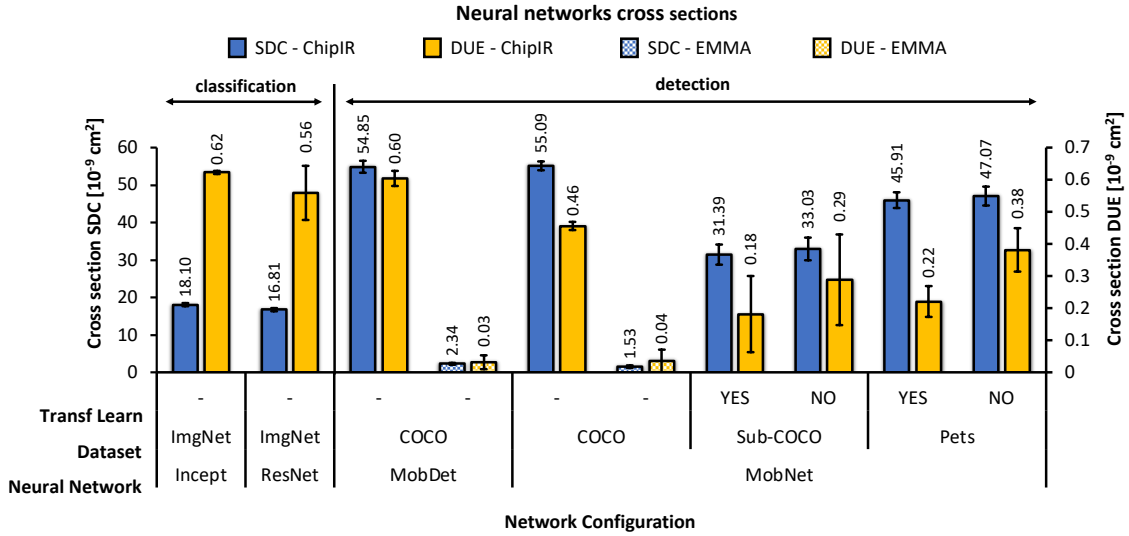
We also retrained MobileNet using two different datasets (a subset of the COCO dataset and a subset of the Oxford-IIIT Pet dataset) with and without applying the transfer learning technique. By retraining the CNN models, we want to evaluate: (1) how the number of object classes supported by the CNN impacts its reliability, since the datasets used for retraining have much less classes than the original COCO dataset and (2) whether transfer learning has a positive effect into the detection resilience.

### 4.2.1 Error rates and Cross Sections

Figure 4.4 plots both SDC cross sections (left Y-axis) and DUE cross sections (right Y-axis) for the eight CNN configurations obtained during experiments with high-energy neutrons at Chip IR facility and for the two CNN configurations tested at EMMA (MobDet and MobNet). Other configurations could not be tested at EMMA due to the low error rate and none of the NNs could be tested at TENIS because the error rate was too high. The four configurations on the left side are the original models that Google provide on the Coral Edge TPU official website. The ones disposed on the right side are retrained versions, with and without transfer learning, of the MobileNet with different datasets.

At ChipIR, the lowest SDC cross section is measured for ResNet. Assuming a flux of $13n/cm^2/h$ for atmospheric neutrons at sea level, the $\sim 17 \times 10^{-9}\ cm^2$ cross section of the ResNet neural network translates to about 270 FIT. This result is very similar to the $10^2$ FIT (same order of magnitude) measured for the NeuroShield with a similar neural network in (BLOWER et al., 2021).

Figure 4.4: Cross sections for the eight CNN configurations that were exposed to high-energy neutrons at Chip IR facility and to thermal neutrons at EMMA facility




At EMMA, MobileDet is confirmed to be 50% more likely to experience SDCs than MobileNet. Although the trend is the same, the SDC cross sections are, on average, 25 times smaller than the corresponding values obtained for these two network configurations when exposed to high-energy neutrons.

From the results plotted in Figure 4.4, we observe that detection networks are less tolerant than classification ones. This can be justified because, although the classification models are larger and, possibly execute more operations, the detection output is much more complex. For the classification task, the output values simply represent the probability of each object class while, in the detection task, the output is composed of six values for each possibly detected object: its class, its probability and its position (x, y, width, height). The position elements are much more sensitive to the effects of faults and, thus, detection CNNs will have higher error rates. This behavior is in accordance with what has been observed in GPUs architectures (Santos et al., 2019).

Transfer learning (TL) does not seem to have a significant impact on the CNN cross sections. This technique has shown to decrease the SDC cross section in only 2-5% when compared to the analogous configuration without TL. However, the training process tend to converge much faster with this strategy and, in our case, it reduced the learning time of the CNNs in around 50%. So, TL is a good solution when a quick re-training of the NN is needed, as it is fast but does not impact the error rate.

Our results also confirm that the retraining of MobileNet with the COCO subset (14 classes) lowers the cross sections when compared to the original model trained with

the total amount of 90 classes of the original COCO dataset. The same network but trained with the Pets dataset (2 classes) have higher cross sections than the one trained with Sub-COCO, but still smaller than the one obtained for the original with the entire COCO dataset. This trend evinces that, with less classes to be considered, the detection process gets simpler and the cross section is reduced. Therefore, the training of CNNs should target the real application needs and include classes of object that are really relevant to the context of the application.

ResNet and Inception, which are CNNs that perform image classification (not detection), have the highest DUEs cross sections. This might be related to the size of the model for these two networks which are 5 to 7 times larger than the MobileNet model. With larger models, the communication between the TPU and the Raspberry Pi increases, as well as the demand for control and hardware resources in general grows, making DUEs more likely to occur. Apart from the retrained networks, which have the lowest value for DUE cross sections, the overall DUE rate is similar among the other CNNs which enforces the fact that DUEs are mostly related to the hardware attributes rather than the algorithm.

Finally, Table 4.2 presents the FIT rates for the CNNs tested at ChipIR. Note that the FIT rates are considered very high. By way of comparison, ISO-26262 states that the error rate in safety-critical application should not exceed 10 FIT (DONGARRA et al., 2015). Even though the TPU is definitely not designed for this kind of application, it is worth it to mentioning that especially the SDC rates are surprisingly high.

Table 4.2: FIT rates for the CNN configurations that were exposed to atmospheric neutrons at Chip IR

| **Network architecture** | **Dataset** | **Transfer learning** | **FIT SDCs** | **FIT DUEs** |
|---|---|---|---|---|
| Inception | ImageNet | - | 235.27 | 8.10 |
| ResNet | ImageNet | - | 218.52 | 7.27 |
| MobileDet | COCO | - | 713.07 | 7.85 |
| MobileNet | COCO | - | 716.15 | 5.92 |
| | Sub-COCO | YES | 408.13 | 2.35 |
| | | NO | 429.44 | 3.74 |
| | Pets | YES | 596.86 | 2.86 |
| | | NO | 611.97 | 4.96 |

### 4.2.2 Critical Errors

As already mentioned, not all SDCs are critical for neural networks execution. As shown in 4.5, SDCs are considered critical when they affect the detection/classification outcome by altering:

- the class of the detected objects;
- the number of detected objects;
- the object position in such a way that the intersection between the expected bounding box and the corrupted one becomes less than 50%.

Figure 4.5: Categories of critical errors

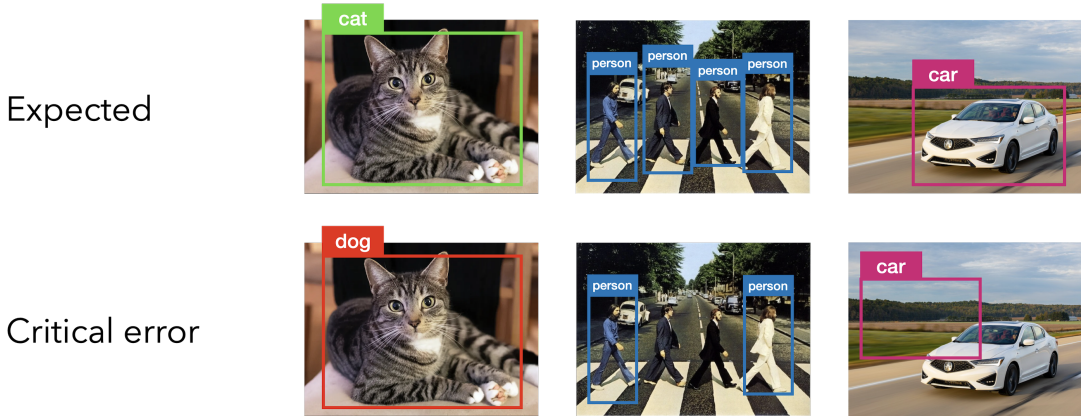


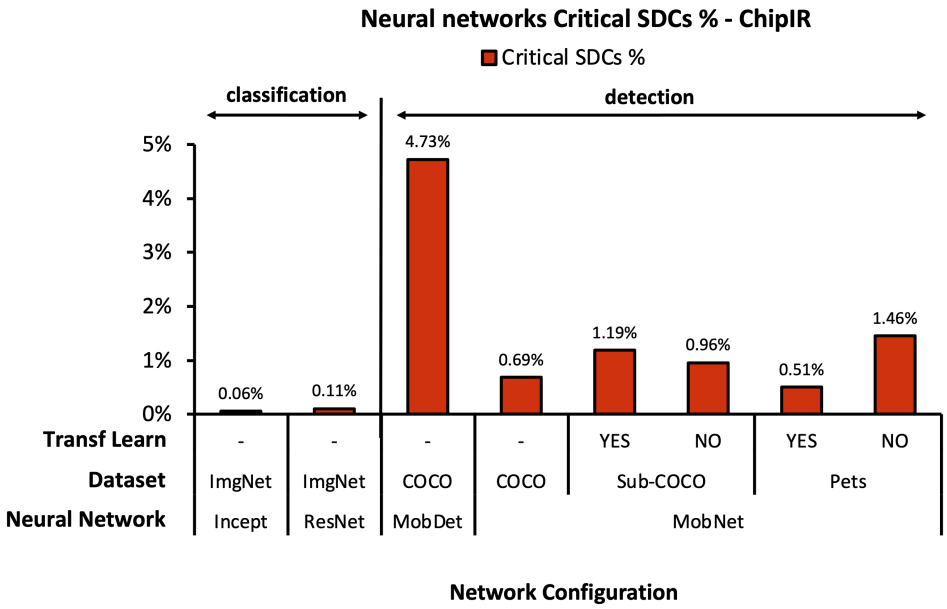

Figure 4.6 shows, for the configurations presented in Figure 4.4, the percentage of SDCs that critically affect the classification/detection outcome.

The great majority of SDCs do not modify classification/detection. In the worst case, only 5% of the errors were considered critical for MobileDet. The average for all evaluated CNNs was only 1.21%. This is again contrasts with results obtained for GPU architectures, in another study by our research group (Santos et al., 2019), in which up to 60% of SDCs in CNNs are classified as critical.

It is clear that SDCs in MobileDet tend to be way more critical than the other network architectures. Comparing it to the MobileNet network architecture, which also performs the detection task, MobileDet has less model parameters, a 13% larger input size and a 50% smaller output. The fact that MobileDet has half of the number of output elements makes each one of them twice more significant fot the detection outcome and, therefore, the corruption of a single output value tends to be more critical in MobileDet than MobileNet.

Transfer learning does not seem to have a consistent impact on the criticality of the

Figure 4.6: Percentage of SDCs that critically affected the classification/detection outcome of the CNN configurations that were exposed to high-energy neutrons at Chip IR facility




SDCs. In the case where MobileNet is retrained with the Pets dataset, the application of this technique has shown to decrease the number of critical errors by almost 3 times. On the other hand, when trained with COCO subset, it makes the NN 20% more susceptible to critical SDCs. Further studies are necessary to understand the reasons for this opposite trend. The differences, though, are not very high.

Naturally, SDCs in classification NNs are considerably less critical since only a few values, the highest ones, out of 1,000 output values are indeed relevant to the outcome of the classification process. Therefore, although the SDCs are propagated to the network raw output, most of them do not influence the classification result, as confirmed by our data plot in Figure 4.6.

# 5 CONCLUSION AND FUTURE WORK

In this work, we have deeply evaluated the reliability of Google Tensor Processing Units through radiation beam experiments with two types of particles: high-energy and thermal neutrons.

First, we have understood how neutrons impact the execution of two types of convolutions (standard 2D and depthwise 3D convolutions), which are the core atomic operations of the TPU, with increasing input matrices size. Besides the not surprising linear dependence between the input size and the cross section values, we have seen that most neutrons corrupt only one element of the output matrix and the corrupted value is very close to the expected value. These characteristics of the fault model are very promising attributes for the reliability of the TPU architecture. With single corrupted elements and small error magnitude, the amount of critical SDCs tend to be drastically reduced as demonstrated in Section 4.2.2.

Then, we have executed eight different configurations of convolutional neural networks on the irradiated TPU. We have seen that networks that perform the detection task have a much higher error rate than those that perform only classification. We have also shown that the dataset reduction has a positive impact on the CNN execution reliability. With less object classes in the dataset to be considered by the CNN, the detection becomes simpler and the CNN gets more robust. Besides that, transfer learning has been shown to significantly reduce training time without comprosing the robustness of the neural networks. Furthermore, the vast majority of errors are not critical for the CNNs execution, which is strictly related to the fault model observed for convolutions.

In general, the DUE cross sections are about 2 orders of magnitude lower than SDC cross sections, which indicates that TPU implements a very reliable interface with the host device to which it is connected. Finally, the TPU seems more prone to being corrupted by high-energy neutrons than by thermal neutrons.

As a future work, we plan to re-train the neural networks using the information we gathered about the convolution error model. The idea is to reduce the number of objects that can be detected and spread them in the whole representation span of the UINT8 data type.

## REFERENCES

ABADI, M. et al. **TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems**. 2015. Software available from tensorflow.org. Available from Internet: <https://www.tensorflow.org/>.

BASSO, P. M. et al. Impact of tensor cores and mixed precision on the reliability of matrix multiplication in gpus. **IEEE Transactions on Nuclear Science**, v. 67, n. 7, p. 1560–1565, 2020.

BEAUCOUR, J. et al. Grenoble large scale facilities for advanced characterisation of microelectronics devices. In: **2015 15th European Conference on Radiation and Its Effects on Components and Systems (RADECS)**. [S.l.: s.n.], 2015. p. 312–316.

BLOWER, S. et al. Evaluating and mitigating neutrons effects on cots edgeai accelerators. **IEEE Transactions on Nuclear Science**, v. 68, n. 8, p. 1719–1726, 2021.

BOSIO, A. et al. A reliability analysis of a deep neural network. In: **2019 IEEE Latin American Test Symposium (LATS)**. [S.l.: s.n.], 2019. p. 157–162.

BREWER, R. M. et al. The impact of proton-induced single events on image classification in a neuromorphic computing architecture. **IEEE Transactions on Nuclear Science**, v. 67, n. 1, p. 108–115, 2020.

CAZZANIGA, C.; FROST, C. D. Progress of the scientific commissioning of a fast neutron beamline for chip irradiation. **Journal of Physics: Conference Series**, IOP Publishing, v. 1021, p. 012037, may 2018. Available from Internet: <https://doi.org/10.1088/1742-6596/1021/1/012037>.

CAZZANIGA, C. et al. Dosimetry of thermal neutron beamlines at a pulsed spallation source for application to the irradiation of microelectronics. **IEEE Transactions on Nuclear Science**, v. 68, n. 5, p. 921–927, 2021.

CHIESA, D. et al. Measurement of the neutron flux at spallation sources using multi-foil activation. **Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment**, v. 902, p. 14–24, 03 2018.

DONGARRA, J. et al. **ISO26262 Standard**. 2015. Available from Internet: <https://www.iso.org/obp/ui/\#iso:std:iso:26262:-1:ed-1:v1:en>.

Fernandes dos Santos, F. et al. Reliability evaluation of mixed-precision architectures. In: **2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)**. [S.l.: s.n.], 2019. p. 238–249. ISSN 1530-0897.

FRATIN, V. et al. Code-dependent and architecture-dependent reliability behaviors. In: **2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)**. [S.l.: s.n.], 2018. p. 13–26.

GAMBARDELLA, G. et al. Efficient error-tolerant quantized neural network accelerators. **2019 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)**, IEEE, p. 131–136, Oct 2019. Available from Internet: <http://dx.doi.org/10.1109/DFT.2019.8875314>.

GUPTA, S. et al. Deep learning with limited numerical precision. In: BACH, F.; BLEI, D. (Ed.). **Proceedings of the 32nd International Conference on Machine Learning**. Lille, France: PMLR, 2015. (Proceedings of Machine Learning Research, v. 37), p. 1737–1746. Available from Internet: <https://proceedings.mlr.press/v37/gupta15.html>.

Hanif, M. A. et al. Error resilience analysis for systematically employing approximate computing in convolutional neural networks. In: **2018 Design, Automation Test in Europe Conference Exhibition (DATE)**. [S.l.: s.n.], 2018. p. 913–916.

HE, K. et al. Deep residual learning for image recognition. In: **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2016. p. 770–778.

IBM Cloud Education. **Convolutional Neural Networks**. [S.l.]: IBM, 2020. <https://www.ibm.com/cloud/learn/convolutional-neural-networks>. Accessed: 2022-04-21.

LIBANO, F. et al. Understanding the impact of quantization, accuracy, and radiation on the reliability of convolutional neural networks on fpgas. **IEEE Transactions on Nuclear Science**, v. 67, n. 7, p. 1478–1484, 2020.

LIBANO, F. et al. On the reliability of linear regression and pattern recognition feedforward artificial neural networks in fpgas. **IEEE Transactions on Nuclear Science**, v. 65, n. 1, p. 288–295, 2018.

LIN, T. et al. Microsoft COCO: common objects in context. **CoRR**, abs/1405.0312, 2014. Available from Internet: <http://arxiv.org/abs/1405.0312>.

MRAZEK, V. et al. Design of power-efficient approximate multipliers for approximate artificial neural networks. In: **2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)**. [S.l.: s.n.], 2016. p. 572–578.

OLIVEIRA, D. et al. Thermal neutrons: a possible threat for supercomputer reliability. **The Journal of Supercomputing**, v. 77, n. 2, p. 1612–1634, 2021. Available from Internet: <https://doi.org/10.1007/s11227-020-03324-9>.

PANDEY, A. **Depth-wise Convolution and Depth-wise Separable Convolution**. 2018. <https://medium.com/@zurister/depth-wise-convolution-and-depth-wise-separable-convolution-37346565d4ec>. Accessed: 2022-04-23.

PARKHI, O. M. et al. Cats and dogs. In: **IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2012.

Q-ENGINEERING. **Google Coral Edge TPU explained in depth.** 2019. Accessed: 2021-08-27. Available from Internet: <https://qengineering.eu/google-corals-tpu-explained.html>.

RECH, P. et al. An efficient and experimentally tuned software-based hardening strategy for matrix multiplication on gpus. **IEEE Transactions on Nuclear Science**, v. 60, n. 4, p. 2797–2804, 2013.

REDMON, J. et al. You only look once: Unified, real-time object detection. **CoRR**, abs/1506.02640, 2015. Available from Internet: <http://arxiv.org/abs/1506.02640>.

RUSSAKOVSKY, O. et al. ImageNet Large Scale Visual Recognition Challenge. **International Journal of Computer Vision (IJCV)**, v. 115, n. 3, p. 211–252, 2015.

SANDLER, M. et al. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In: **2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2018. p. 4510–4520.

Santos, F. F. d. et al. Analyzing and increasing the reliability of convolutional neural networks on gpus. **IEEE Transactions on Reliability**, v. 68, n. 2, p. 663–677, 2019.

Sarwar, S. S. et al. Energy efficient neural computing: A study of cross-layer approximations. **IEEE Journal on Emerging and Selected Topics in Circuits and Systems**, v. 8, n. 4, p. 796–809, 2018.

SLAYMAN, C. JEDEC Standards on Measurement and Reporting of Alpha Particle and Terrestrial Cosmic Ray Induced Soft Errors. [S.l.]: IEEE, 2010. v. 41, p. 55–76. ISBN 978-1-4419-6992-7.

SZEGEDY, C. et al. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In: **Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence**. [S.l.]: AAAI Press, 2017. (AAAI'17), p. 4278–4284.

WEULERSSE, C. et al. Contribution of thermal neutrons to soft error rate. **IEEE Transactions on Nuclear Science**, v. 65, n. 8, p. 1851–1857, 2018.

XIONG, Y. et al. MobileDets: Searching for Object Detection Architectures for Mobile Accelerators. **CoRR**, abs/2004.14525, 2020. Available from Internet: <https://arxiv.org/abs/2004.14525>.